

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

Кафедра системного програмування і спеціалізованих комп'ютерних систем

«До захисту допущено»

Завідувач кафедри

(підпис) Тарасенко В.П.
(ініціали, прізвище)

“ ____ ” червня 2019 р.

**Дипломний проект
на здобуття ступеня бакалавра**

з напрямку підготовки **6.050102 «Комп'ютерна інженерія»**

на тему: Веб-орієнтований додаток для компресії HTML файлів

Виконала: студентка IV курсу, групи KB-52

Таценко Мирослава Іванівна

(підпис)

Керівник доц.каф.СПСКС, к.т.н. Клятченко Я.М.

(підпис)

Консультант з нормоконтролю, доц.каф.СПСКС, к.т.н. Клятченко Я.М.

(підпис)

Рецензент _____

(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали)

(підпис)

Засвідчую, що у цьому дипломному
проекті немає запозичень з праць інших
авторів без відповідних посилань.

Студент _____
(підпис)

Київ – 2019 року

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

Кафедра системного програмування і спеціалізованих комп'ютерних систем

Рівень вищої освіти – перший (бакалаврський)

Напрямок підготовки 6.050102 «Комп'ютерна інженерія»

ЗАТВЕРДЖУЮ

Завідувач кафедри

(підпис) Тарасенко В.П.
(ініціали, прізвище)

«__» червня 2019 р.

**ЗАВДАННЯ
на дипломний проект студентки
Таценко Мирослави Іванівни**

1. Тема проекту веб-орієнтований додаток для компресії HTML файлів,
керівник проекту доц.каф.СПСКС, к.т.н. Клятченко Я.М. ,

затверджені наказом по університету від «22» травня 2019 р. №1330-С

2. Термін подання студентом проекту: дивись технічне завдання.

3. Вихідні дані до проекту: титульна сторінка, програма, технічне завдання (4 сторінки), пояснювальна записка (50 сторінок), п'ять додатків.

4. Зміст пояснювальної записки: перелік скорочень, умовних позначень та термінів; вступ; аналіз існуючих веб-орієнтованих додатків для компресії HTML файлів та обґрунтування теми дипломного проекту; аналіз мов програмування та технологій розроблення веб-орієнтованих систем; обґрунтування вибору засобів реалізації; розробка веб-орієнтованого додатку для компресії HTML файлів; висновки; список використаної літератури.

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслеників, плакатів, презентацій тощо): презентація; структурна схема проекту; структурна схема взаємодії модулів проекту; схема алгоритму компресії HTML файлу; схема алгоритму роботи з «білими» символами.

6. Консультанти розділів проекту*

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання

Календарний план

№ з/п	Назва етапів виконання дипломного проекту	Термін виконання етапів проекту	Примітка
1.	Вивчення літератури за тематикою проекту	15.04.2019	
2.	Розроблення та узгодження технічного завдання	30.04.2019	
3.	Аналіз існуючих рішень	05.05.2019	
4.	Підготовка матеріалів першого розділу дипломного проекту	10.05.2019	
5.	Підготовка матеріалів другого розділу дипломного проекту	18.05.2019	
6.	Підготовка графічної частини дипломного проекту	20.05.2019	
7.	Оформлення документації дипломного проекту	25.05.2019	
8.	Попередній огляд матеріалів диплому на кафедрі	30.05.2019	

Студент

(підпис)

(ініціали, прізвище)

Керівник проекту

(підпис)

(ініціали, прізвище)

* Консультантом не може бути зазначено керівника дипломного проекту.

АНОТАЦІЯ

Об'єкт розробки – веб-орієнтований додаток для компресії HTML файлів.

Компресія файлів - це процес "упаковки" файлу (або файлів) для використання меншого дискового простору. Програмне забезпечення стиснення дозволяє приймати багато файлів і стискати їх в один файл, який менше, ніж комбінований розмір оригіналів. Компресія безпосередньо HTML файлів допоможе не лише заощадити місце але й зменшити завантаження веб-сторінок.

- проведено аналіз існуючих веб-орієнтованих рішень для компресії HTML файлів;
- сформульовані вимоги до веб-орієнтованого додатку для компресії HTML файлів;
- проведено аналіз допоміжного програмного забезпечення для стверення веб-орієнтованого додатку для компресії HTML файлів;
- розроблено веб-орієнтований додаток для компресії HTML файлів.

Упровадження даного додатку у веб-мережу дасть змогу полегшити і пришвидчити роботу розробникам.

Ключові слова:

ВЕБ-ОРІЄНТОВАНИЙ ДОДАТОК ДЛЯ КОМПРЕСІЇ HTML ФАЙЛІВ,
КОМПРЕСІЯ, HTML.

SUMMARY

The object of the development is a web-based application for compressing HTML files.

Compression of files is a process of "packing" a file (or systems) to use less disk space. The compression software will allow many files and styles of their single file to be smaller than the combined original size. Compressing directly HTML files helps to not only delay the site, but also reduce downloading of web pages.

- An analysis of existing web-based solutions for compressing HTML files;
- Leo formulates the requirements for a web-based application for compressing HTML files;
- The analysis of auxiliary software for approval of the web-based application for compressing HTML files;
- A web application for compressing HTML files is developed.

Implementation of this application in the web-site will allow you to move and develop the work of the developer.

Keywords:

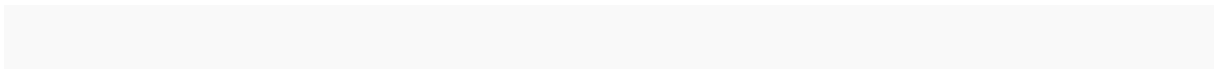
WEB-BASED APPLICATION FOR HTML FILES COMPRESSION,
COMPRESSION, HTML.

[illegible]

Поз.	Формат	ПОЗНАЧЕННЯ	НАЙМЕНУВАННЯ	Кількість аркушів	№ прим.	Примітки
	A4	ІАЛЦ.467200.006 Д2	Веб-орієнтований додаток	1		
			для компресії HTML			
			файлів			
			Схема взаємодії модулів			
			програми			
	A4	ІАЛЦ.467200.007 Д3	Веб-орієнтований додаток	1		
			для компресії HTML			
			файлів			
			Схема алгоритму компресії			
			HTML файлів			
	A4	ІАЛЦ.467200.008 Д4	Веб-орієнтований додаток	1		
			для компресії HTML			
			файлів			
			Схема алгоритму обробки			
			«білих» символів			
		Диск CD-ROM	Текст ПЗ. Тексти програм.	1		
			Графічний матеріал.			
ІАЛЦ.467200.001 ОА						Арк.
Змін.	Арк.	№ докум.	Підпис	Дата	2	

ЗМІСТ

1. НАЙМЕНУВАННЯ ТА ГАЛУЗЬ РОЗРОБКИ.....	2
2. ПІДСТАВА ДЛЯ РОЗРОБКИ	2
3. ЦІЛЬ І ПРИЗНАЧЕННЯ РОБОТИ	2
4. ДЖЕРЕЛА РОЗРОБКИ.....	2
5. ТЕХНІЧНІ ВИМОГИ.....	2
5.1. Вимоги до програмного продукту, що розробляється	2
5.2. Вимоги до апаратного забезпечення.....	3
5.3. Вимоги до програмного та апаратного забезпечення користувача	3
6. ЕТАПИ РОЗРОБКИ	4



					ІАЛЦ. 467200.002 ТЗ					
Зм	Лист	№ докум.	Підп.	Дата	Веб-орієнтований додаток для компресії HTML файлів			Лім.	Лист	Листів
Розроб.		Таценко								
Перев.		Клятченко							1	4
								КПІ ім. Ігоря Сікорського, ФПМ, КВ-52		
Н. контр.		Клятченко								
Затв.		Тарасенко								
					Технічне завдання					

1. НАЙМЕНУВАННЯ ТА ГАЛУЗЬ РОЗРОБКИ

Назва розробки: «Веб-орієнтований додаток для компресії HTML файлів».
Галузь застосування: веб-мережа інтернет.

2. ПІДСТАВА ДЛЯ РОЗРОБКИ

Підставою для розробки є завдання на виконання роботи ступеня «бакалавр комп'ютерної інженерії», затверджене кафедрою системного програмування і спеціалізованих комп'ютерних систем Національного технічного університету України «Київський Політехнічний Інститут імені Ігоря Сікорського».

3. МЕТА І ПРИЗНАЧЕННЯ РОБОТИ

Метою даного проекту є створення веб-орієнтованого додатку для компресії HTML файлів.

4. ДЖЕРЕЛА РОЗРОБКИ

Джерелом інформації є технічна та науково-технічна література, технічна документація, публікації у періодичних виданнях та електронні статті у мережі Інтернет.

5. ТЕХНІЧНІ ВИМОГИ

5.1. Вимоги до програмного продукту, що розробляється

- сумісність з будь-яким веб-браузером;
- можливість управління процесом компресії;
- можливість компресії HTML файлу будь-якого розміру;
- можливість перегляду відсоткового відношення компресії;
- простий і зрозумілий інтерфейс.

					ІАЛЦ.467200.002 ТЗ	Лист 2
Зм	Лист	№ докум.	Підп.	Дата		

5.2. Вимоги до апаратного забезпечення

- Процесор: 2,4-ядерний;
- Оперативна пам'ять: 5 Гб;
- Наявність доступу до мережі Internet (GPRS, EDGE, 3G, 4G).

5.3. Вимоги до програмного та апаратного забезпечення користувача

- Наявність доступу до мережі Internet (GPRS, EDGE, 3G, 4G);
- Веб-орієнтований додаток.

					ІАЛЦ.467200.002 ТЗ	Лист
						3
Зм	Лист	№ докум.	Підп.	Дата		

6. ЕТАПИ РОЗРОБКИ

№ з/п	Назва етапів виконання дипломного проекту	Термін виконання етапів
1.	Вивчення літератури за тематикою проекту	15.04.2019
2.	Розроблення та узгодження технічного завдання	30.04.2019
3.	Аналіз існуючих рішень	05.05.2019
4.	Підготовка матеріалів першого розділу дипломного проекту	10.05.2019
5.	Підготовка матеріалів другого розділу дипломного проекту	18.05.2019
6.	Підготовка графічної частини дипломного проекту	20.05.2019
7.	Оформлення документації дипломного проекту	25.05.2019
8.	Попередній огляд матеріалів диплому на кафедрі	30.05.2019

Поз.	Формат	ПОЗНАЧЕННЯ	НАЙМЕНУВАННЯ	Кількість аркушів	№ прим.	Примітки
			<u>Документація загальна</u>			
			<u>Новорозроблена</u>			
	A4	ІАЛЦ.467200.004 ПЗ	Веб-орієнтований додаток	50		
			для компресії HTML			
			файлів			
			Пояснювальна записка			
	A4	ІАЛЦ.467200.005 Д1	Веб-орієнтований додаток	1		
			для компресії HTML			
			файлів			
			Структурна схема			
			розташування модулів			
			програми			
	A4	ІАЛЦ.467200.006 Д2	Веб-орієнтований додаток	1		
			для компресії HTML			
			файлів			
			Схема взаємодії модулів			
			програми			
	A4	ІАЛЦ.467200.007 Д3	Веб-орієнтований додаток	1		
			для компресії HTML			
			файлів			
			Схема алгоритму компресії			
			ДП.467200.003 ТП			
Змін.	Арк.	№ докум.	Підпис	Дата	<div> <div>Літ.</div> <div>Аркуш</div> <div>Аркушів</div> </div> <div> <div>1</div> <div>2</div> </div> <div>КПІ ім. Ігоря Сікорського, ФПМ, КВ-52</div>	
Розробив		Гащенко М.І.				
Перевірив		Клятченко Я.М.				
Консульт.						
Н. контроль		Клятченко Я.М.				
Зав. каф.		Тарасенко В.П.				
					<div> <div>Веб-орієнтований додаток для компресії HTML файлів</div> <div>Відомість технічного проекту</div> </div>	

[illegible]

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, ТЕРМІНІВ	3
ВСТУП	4
1. АНАЛІЗ ІСНУЮЧИХ ВЕБ-ОРІЄНТОВАНИХ ДОДАТКІВ ДЛЯ КОМПРЕСІЇ HTML ФАЙЛІВ ТА ОБГРУНТУВАННЯ ТЕМИ ДИПЛОМНОГО ПРОЕКТУ	5
1.1. Що таке веб-орієнтований додаток	5
1.2. Актуальність теми компресії в комп'ютерних технологіях	9
1.3. Аналіз існуючих веб-орієнтованих додатків для компресії HTML файлів	13
1.4. Обґрунтування теми дипломного проекту	16
2. АНАЛІЗ МОВ ПРОГРАМУВАННЯ ТА ТЕХНОЛОГІЙ РОЗРОБЛЕННЯ ВЕБ-ОРІЄНТОВАНИХ СИСТЕМ	19
2.1. Порівняльний аналіз мов програмування Python та Java Script	19
2.2. Аналіз реалізацій веб-архітектур	27
3. ОБГРУНТУВАННЯ ВИБОРУ ЗАСОБІВ РЕАЛІЗАЦІЇ	29
3.1. Веб-платформа Node.js	29
3.2. Вибір шаблонізатора та фреймворка для верстки	32
3.3. Вибір засобів хостингу веб-додатку	35

					ІАЛЦ.467200.004 ПЗ					
Зм	Лист	№ докум.	Підп.	Дата	Веб-орієнтований додаток для компресії HTML файлів			Лім.	Лист	Листів
Розроб.	Таценко									
Перев.	Клятченко								1	49
								КПІ ім. І. Сікорського, ФПМ, КВ-52		
Н. контр.	Тарасенко									
Затв.	Клятченко				Пояснювальна записка					

4. РОЗРОБКА ВЕБ-ОРІЄНТОВАНОГО ДОДАТКУ ДЛЯ КОМПРЕСІЇ HTML ФАЙЛІВ _____	40
4.1. Інтерфейс веб-орієнтованого додатку _____	41
4.2. Структура проекту веб-орієнтованого додатку _____	42
4.3. Алгоритм компресії HTML файлів _____	44
ВИСНОВОК _____	48
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ _____	49
ДОДАТКИ	

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, ТЕРМІНІВ

ПЗ – програмне забезпечення.

HTML (Hypertext Markup Language) - мова розмітки гіпертекстових документів. Стандартна мова розмітки для створення веб-сторінок і веб-додатків.

WWW (World Wide Web) - збірна колекція файлів.

IBM (International Business Machines Corporation) - американська електронна корпорація, один із найбільших світових виробників усіх видів комп'ютерів і програмного забезпечення.

IP – інтернет протокол мережевого рівня.

TCP – інтернет протокол транспортного рівня.

					ІАЛЦ.467200.004 ПЗ	Лист
						3
Зм	Лист	№ докум.	Підп.	Дата		

ВСТУП

В сучасному світі дуже гостро стоїть проблема компресії. Ми намагаємося заощадити не лише місце, але й час, що на даному етапі розвитку веб-технологій один з найважливіших показників. Навіть від долі секунди витраченої в пусту залежить боротьба за кожного користувача. Сучасний темп розвитку технологій невинно диктує нам, що ми повинні рухатися і надавати користувачам інформацію все з більшою швидкістю.

Для пришвидшення завантаження сторінок використовують компресію HTML файлів. Це не лише економить час завантаження сторінки, а й зменшує місце, яке використовується, на сервері.

					ІАЛЦ.467200.004 ПЗ	Лист
						4
Зм	Лист	№ докум.	Підп.	Дата		

1. АНАЛІЗ ІСНУЮЧИХ ВЕБ-ОРІЄНТОВАНИХ ДОДАТКІВ ДЛЯ КОМПРЕСІЇ HTML ФАЙЛІВ ТА ОБҐРУНТУВАННЯ ТЕМИ ДИПЛОМНОГО ПРОЕКТУ

1.1. Що таке веб-орієнтований додаток

Інтернет, і межі його можливостей, є результатом фундаментального принципу мережевої інженерії: Keep It Simple, Stupid (KISS). Кожен комп'ютер, підключений до Інтернету, здатний виконати кілька дуже простих завдань дуже швидко. Поєднуючи мільйони порівняно простих систем, досягається комплексна функціональність. Інтернет є геніальною мережею комунікацій здебільшого тому, що це так просто.

В основі будь-якої передачі Інтернету (надсилання електронної пошти, перегляд веб-сторінки або завантаження аудіофайлу або відеофайлу) - є Інтернет-протокол (IP). Винайдений у 1974 році Вінтом Серфом і Робертом Каном, IP є схемою зв'язку, яка визначає, як передаються дані по мережах. IP має два ключові стандартизовані елементи, які беруть участь у кожній передачі. Перший загальний метод для розбиття кожної передачі вниз на невеликі шматки даних, відомі як "пакети". А також другий елемент - єдина глобальна система адресації. IP надає кожному комп'ютеру, підключеному до Інтернету, унікальну адресу, а також загальне визначення пакетів даних, які можуть бути доставлені на ці адреси.

Іншими словами, Інтернет-протокол зводиться до двох простих правил:

1. Кожен комп'ютер, підключений до Інтернету, повинен бути доступним через числову адресу конкретної форми: чотири восьмибітові числа, розділені періодами. Таку адресу називають "адресою Інтернет-протоколу" або "IP-адресою". Наприклад, IP-адреса для домашньої сторінки Google - 216.239.51.100. Що стосується більшості комп'ютерів Інтернету, то IP-адреса - це все, що вам дійсно потрібно.

2. Кожен комп'ютер, підключений до Інтернету, повинен мати можливість приймати пакети з заголовком від 24 до 32 байт і розміром пакета до 576 байтів. Заголовок містить інформацію про початок і адресу призначення кожного пакета і загальний розмір пакета.

Оскільки IP-протокол є таким простим, є багато корисних функцій, не включених до протоколу. Можливо, найважливішою з цих ключових особливостей є "гарантована доставка". Використовуючи "чистий" IP, комп'ютер спочатку розбиває повідомлення, яке надсилається на невеликі пакети, кожен з яких позначається адресою машини призначення. Комп'ютер потім передає ці пакети до наступної підключеної Інтернет-машини, яка переглядає адресу призначення, а потім передає її на наступну підключену Інтернет-машину, яка переглядає адресу призначення і передає її разом, і так далі, поки пакети не досягнуть машини призначення. Таким чином, IP є послугою комунікації "найкращих зусиль", що означає, що він робить все можливе для доставки пакетів відправника до визначеного пункту призначення, але він не може давати жодних гарантій.

Сам по собі IP не може гарантувати, що пакети прибули у правильному порядку або навіть прийшли взагалі. Це робота іншого протоколу: TCP (Transmission Control Protocol). TCP розташований "зверху" IP і гарантує, що всі пакети, відправлені з однієї машини на іншу, приймаються і збираються в правильному порядку. Якщо будь-який з пакетів опускається під час передачі, машина призначення використовує TCP для запиту, це робиться для того, щоб відправляюча машина повторно відправила втрачені пакети, і визнає їх, коли вони надходять. Завдання TCP полягає в тому, щоб переконатися, що передача отримана в повному обсязі, і сповістити відправника, що всі дані прийшли нормально.

Веб-сайт (або веб-орієнтований додаток) - це збірна колекція файлів World Wide Web (WWW), що містить початковий файл, який називається домашньою сторінкою. Компанія або людина повідомляє вам, як дістатися до

їхнього веб-сайту, надавши вам адресу своєї домашньої сторінки. За допомогою домашньої сторінки можна отримати всі інші сторінки на сайті. Наприклад, веб-сайт для IBM має адресу домашньої сторінки <http://www.ibm.com>. Адреса домашньої сторінки фактично включає в себе певне ім'я файлу, як `index.html`, але, як у випадку IBM, коли стандартне ім'я за замовчуванням налаштовано, користувачам не потрібно вводити ім'я файлу. Адреса домашньої сторінки IBM приводить до тисяч сторінок. Але веб-сайт також може бути лише на декількох сторінках.

Оскільки сайт передбачає географічне місце, веб-сайт можна сплутати з веб-сервером. Сервер - це комп'ютер, який містить файли для одного або декількох сайтів. Дуже великий веб-сайт може поширюватися на декілька серверів у різних географічних місцях. IBM є гарним прикладом. Її веб-сайт складається з тисяч файлів, розкиданих на багатьох серверах у світових локаціях. Але більш типовим прикладом є, ймовірно, сайт, який ви шукаєте, whatis.com. Ми перебуваємо на сервері провайдера комерційного простору з рядом інших сайтів, які не мають нічого спільного з глосаріями в Інтернеті.

Синонімом і менш часто використовуваним терміном для веб-сайту є "веб-присутності". Цей термін, здається, краще висловлює думку про те, що сайт не прив'язаний до певного географічного розташування, а є "десь у кіберпросторі". Проте "веб-сайт", здається, використовується набагато частіше.

Ви можете мати кілька веб-сайтів, які перехресно посилаються на файли на сайтах один одного або навіть спільно використовують однакові файли.

Веб-сайт повинен бути візуально привабливим, полірованим і професійним. При розробці важливо пам'ятати, що це відображає представлені компанію, продукти та послуги. Веб-сайт може бути першим і єдиним враженням, яке потенційний клієнт отримує від компанії.

Привабливий веб-сайт набагато частіше створюватиме позитивне враження та зберігатиме відвідувачів на веб-сайті після їх надходження. Оскільки великі та малі підприємства продовжують розміщувати в Інтернеті

					ІАЛЦ.467200.004 ПЗ	Лист 7
Зм	Лист	№ докум.	Підп.	Дата		

свої сторінки, задача розробника полягає в залученні та збереженні уваги користувачів.

Ознаки хорошого веб-сайту

1. Гарне використання кольору - відповідна кольорова схема буде містити 2 або 3 основних кольорів, які добре поєднуються і створюють належний настрій або тон для бізнесу. Погана практика перебільшувати з кольором, оскільки він може відволікти від написаного вмісту.
2. Текст, який легко читається - найпростіше читається чорний текст на білому тлі, але багато інших комбінацій кольорів є прийнятними, якщо контраст знаходиться в межах відповідного діапазону. Використовуйте шрифти, які легко читаються і знаходяться на більшості сучасних комп'ютерних систем. Зберігайте достатній розмір шрифту для тексту абзацу.
3. Графіка важлива - оскільки графічний дизайн надає візуальне різноманіття і урізномбарвлює нудні сторінки тексту. Проте не варто зловживати надмірним використанням графіки їх, інтерфейс має збігатися змістом та контекстом із задумкою веб-сайту. Не потрібно перевантажувати жодної сторінки більш ніж на 3 або 4 зображення.
4. Якісні фотографії - простий спосіб підвищити візуальну привабливість полягає у використанні високоякісної фотографії. Високоякісні зображення продуктів особливо важливі для інтернет-магазинів.
5. Простота – потрібно розробляти інтерфейс веб-сайту максимально простим і забезпечувати достатній простір. Не потрібно перевантажувати веб-сайт занадто складним дизайном, анімацією чи іншими ефектами.

1.2. Актуальність теми компресії в комп'ютерних технологіях

Стиснення - це простий, ефективний спосіб збереження пропускної здатності та прискорення роботи вашого сайту. Одним з найпопулярніших типів компресії на стороні серверу є gzip компресія. Вона працює на основі кодування.

Коли відбувається, наприклад оновлення сторінки, запитується файл, такого типу як `http://www.yahoo.com/index.html`, браузер переходить до веб-сервера (рисунок 1.1). Запит відбувається так:

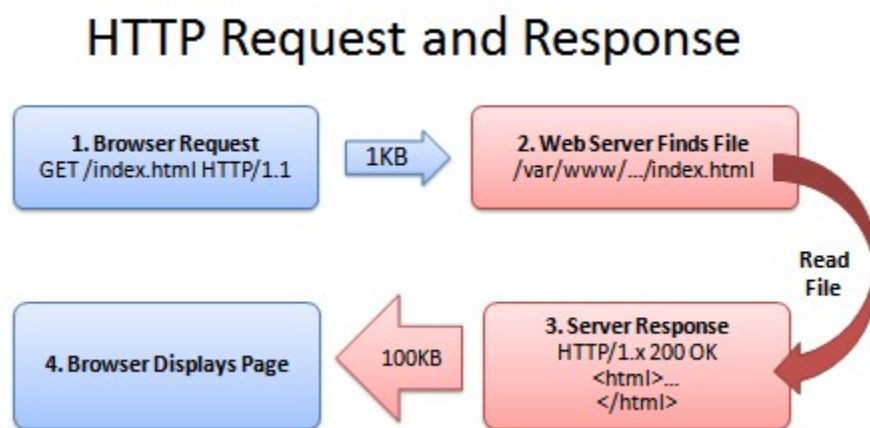


Рисунок 1.1

Браузер: GET /index.html

Сервер: Шукає де лежить index.html ...

Сервер: Знаходить і віддає відповідь (якщо все пройшло доюре, то 200 OK), і я надсилає файл.

Браузер: Починає довго завантажувати файл який важить більше 100 Кб.

Така система отримання відповіді працює, але це не так ефективно. 100 Кб тексту це достатньо багато. Кожен тег `<html>`, `<table>` і `<div>` має закритий тег, який майже не змінюється. Слова повторюються в усьому документі.

Якщо ми можемо надіслати файл .zip у веб-переглядачі (index.html.zip) замість простого старого index.html, ми заощадимо на пропускній здатності та часу завантаження. Браузер може завантажити завантажений файл, витягти

його, а потім показати його користувачеві, який має гарний настрій, оскільки сторінка швидко завантажується.

Ще люди не змогли придумати алгоритм, який буде компресувати краще за gzip. Але є деяка проблема, даний спосіб кодування бере весь HTML файл і не змінюючи його кодує. Такий спосіб звичайно ефективний, проте достатньо велику частину файлу HTML взагалі не потрібно було компресувати, так як вона непотрібна.

Для того щоб зрозуміти наскільки компресія важлива і стоїть на передовому місці в сучасних технологіях поговоримо про нова нещодавно винайдену технологію стиснення даних для пришвидшення роботи комп'ютерних програм.

Стиснення даних використовує надлишкові дані, щоб звільнити ємність пам'яті, підвищити швидкість обчислень і забезпечити інші пільги. У сучасних комп'ютерних системах доступ до основної пам'яті є дуже дорогим порівняно з фактичними обчисленнями. Через це використання стиснення даних в пам'яті сприяє підвищенню продуктивності, оскільки зменшує частоту і обсяг даних, які програми повинні витягувати з основної пам'яті.

Пам'ять у сучасних комп'ютерах управляє та передає дані у фіксованих розмірах, на яких повинні працювати традиційні методи стиснення. Програмне забезпечення, однак, не зберігає свої дані у фіксованих розмірах. Замість цього він використовує "об'єкти", структури даних, які містять різні типи даних і мають змінні розміри. Тому традиційні апаратні методи стиснення обробляють об'єкти погано.

У роботі, представленій на Міжнародній конференції АСМ з архітектурної підтримки мов програмування та операційних систем, дослідники МІТ описують перший підхід до стиснення об'єктів по всій ієрархії пам'яті. Це зменшує використання пам'яті, підвищуючи продуктивність та ефективність.

Програмісти могли б скористатися цією методикою при програмуванні на будь-якій сучасній мові програмування (наприклад, Java, Python і Go), яка

зберігає і керує даними в об'єктах, не змінюючи їх код. Зрештою, споживачі побачать комп'ютери, які можуть працювати набагато швидше або можуть запускати багато інших програм з однаковою швидкістю. Оскільки кожна програма споживає менше пам'яті, вона працює швидше, тому пристрій може підтримувати більше додатків у межах виділеної пам'яті.

В експериментах з використанням модифікованої віртуальної машини Java, техніка стискала вдвічі більше даних і зменшила наполовину використання пам'яті порівняно з традиційними методами, заснованими на кеші.

«Мотивація полягала в тому, щоб створити нову ієрархію пам'яті, яка могла б робити об'єктне стиснення, а не стиснення кеш-ліній, адже саме сучасні мови програмування керують даними», - говорить перший автор Po-An Tsai у лабораторії комп'ютерних наук та штучного інтелекту (CSAIL).

«Усі комп'ютерні системи виграють від цього», - додає співавтор Даніель Санчес, професор інформатики та електротехніки, а також дослідник CSAIL. «Програми стають швидшими, тому що вони перестають бути вузькими через пропускну здатність пам'яті».

Дослідники побудували на своїй попередній роботі, що деструктурує архітектуру пам'яті, щоб безпосередньо маніпулювати об'єктами. Традиційні архітектури зберігають дані в блоках в ієрархії поступово більших і більш повільних пам'яті, що називаються "кешами". Нещодавно доступні блоки піднімаються до менших, більш швидких кеш-пам'яті, тоді як старі блоки переміщуються в більш повільні і більші кеші, зрештою закінчуючись назад в основну пам'ять. Хоча ця організація є гнучкою, вона є дорогою: для доступу до пам'яті кожен кеш повинен шукати адресу серед свого вмісту.

«Оскільки природна одиниця управління даними в сучасних мовах програмування є об'єктами, чому б не просто створити ієрархію пам'яті, яка має справу з об'єктами?» Санчес говорить.

У доповіді, опублікованій в жовтні минулого року, дослідники деталізували систему Notpads, яка зберігає цілі об'єкти, щільно упаковані в ієрархічні рівні, або «колодки». Ці рівні перебувають виключно на ефективних, на чіпі, безпосередньо адресуються пам'яті - без необхідності складних пошуків.

Програми потім безпосередньо посилаються на розташування всіх об'єктів по всій ієрархії колодок. Нещодавно виділені та нещодавно згадані об'єкти, а також об'єкти, на які вони вказують, залишаються на більш швидкому рівні. Коли швидкий рівень заповнюється, він запускає процес «виселення», який зберігає нещодавно згадані об'єкти, але знищує старі об'єкти на більш повільні рівні і переробляє об'єкти, які більше не є корисними, щоб звільнити простір. Потім покажчики оновлюються в кожному об'єкті, щоб вказати на нові місця розташування всіх переміщуваних об'єктів. Таким чином, програми можуть отримати доступ до об'єктів набагато дешевше, ніж пошук через рівні кешу.

Для нової роботи дослідники розробили методику, яка називається «Zippads», яка використовує архітектуру Notpads для стиснення об'єктів. Коли об'єкти спочатку починаються на більш високому рівні, вони не стискаються. Але коли вони виселяються на більш повільні рівні, вони всі стиснуті. Покажчики у всіх об'єктах на різних рівнях потім вказують на ті стислі об'єкти, що дозволяє легко відкликати їх на більш швидкі рівні і здатні зберігатися більш компактно, ніж попередні методи.

Алгоритм стиснення потім ефективно використовує надмірність по об'єктах. Цей метод розкриває більше можливостей стиснення, ніж попередні методи, які були обмежені пошуком надлишковості в кожному блоці фіксованого розміру. Алгоритм спочатку вибирає кілька репрезентативних об'єктів як «базові» об'єкти. Потім у нових об'єктах зберігаються тільки різні дані між цими об'єктами і репрезентативними базовими об'єктами.

1.3. Аналіз існуючих веб-орієнтованих додатків для компресії HTML файлів

Кожен тип даних має деяку надлишковість, тобто втрачене місце в ньому. Якщо текст, як правило, може мати надлишок 60%, цей показник може бути набагато більшим для деяких інших медіа, таких як аудіо та відео. На відміну від тексту, ці інші типи носіїв використовують багато місця для зберігання своїх даних, і необхідність оптимізувати зберігання і відновити простір була очевидною дуже рано. Інженери розробили оптимізований алгоритм стиснення, використовуваний у форматах файлів, призначених для цієї конкретної мети. Алгоритми стиснення, які використовуються для файлів, можуть бути згруповані в дві великі категорії:

1. Стиснення без втрат, де цикл стиснення-розпакування не змінює відновлені дані. Він відповідає (байт до байта) оригіналу. Для зображень gif або png використовуються стиснення без втрат.
2. Стиснення з втратами, коли цикл змінює вихідні дані непомітним способом для користувача.

Деякі формати можуть використовуватися як для стиснення, так і для стиснення з втратами, як, наприклад, webp, і зазвичай алгоритм з втратами може бути налаштований для стиснення більш-менш, що, звичайно, призводить до меншої або більшої якості. Для кращої продуктивності веб-сайту він ідеально підходить для максимального стиснення, зберігаючи при цьому прийнятний рівень якості. Для зображень зображення, створене інструментом, може бути недостатньо оптимізованим для Інтернету; Рекомендується використовувати інструменти, які максимально стискають необхідну якість. Існують численні інструменти, які спеціалізуються на цьому.

Алгоритми стиснення з втратами зазвичай є більш ефективними, ніж стиснення без втрат.

Коли хтось входить на ваш сайт, на сервер надсилається запит на доставку запитаного файлу. Чим більші ці файли, тим більше часу потрібно для завантаження. Стиснувши веб-сторінки та CSS, перш ніж надсилати їх у браузер, ви значно скоротите час завантаження, оскільки файли будуть меншими.

Коротше кажучи, gzip знаходить подібні рядки в текстовому файлі і тимчасово замінює ці рядки, що зменшує загальний розмір файлу. Стиснення Gzip дійсно вигідне в веб-середовищі, оскільки HTML, CSS-файли використовують багато повторюваних текстів.

Але що робити з непотрібними повторами та форматуваннями gzip їх не видалить, а всього на всього приховає. Відповідь це передкомпресія HTML файлів. У великих проектах це роблять за допомогою додатково підключених модулів. Але якщо це особистий проект, то існує безліч онлайн веб-орієнтованих додатків для компресії HTML файлів. Проте якщо почати шукати комфортний додаток саме для вас, то можна розчаруватися.

Розглянемо вже існуючі веб-додатки для компресії HTML файлів.

Перший ресурс який ми розглянемо це <https://compressor.andona.click> (рисунок 1.2).

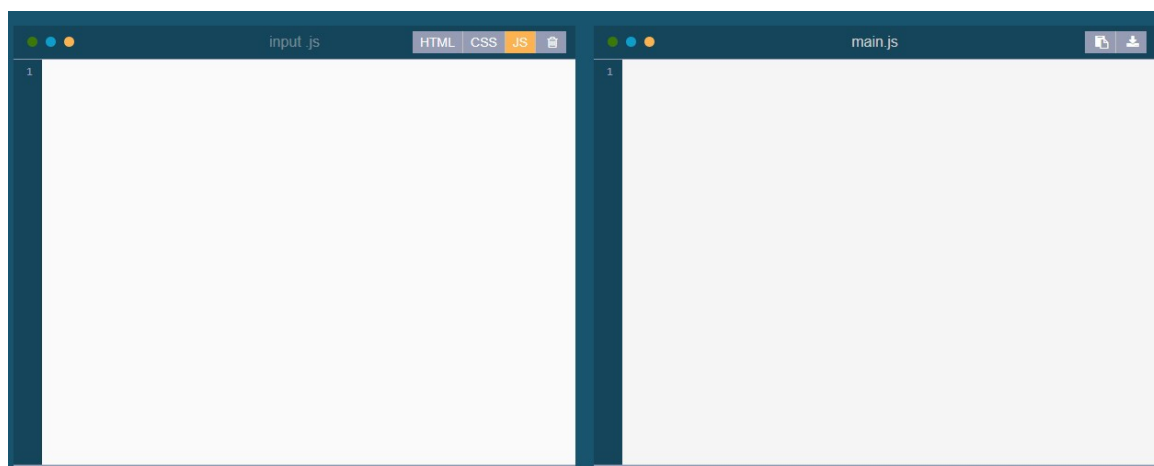


Рисунок 1.2

З першого погляду здається простим та зрозумілим. Нагромадження надмірної графіки майже відсутнє. Плюсом даного веб-сайту також являється

те, що він має можливість здійснювати компресію над різним форматом вхідних даних. Але вже при використанні виникають деякі труднощі. Здавалося б зрозумілий інтерфейс стає незрозумілим. Не досить зрозуміло за що відповідають кнопки, це потрібно в'яснити при безпосередньому натисканні. Веб-сайт виявився незручним.

Наступний веб-сайт для компресії HTML файлів це - <https://htmlcompressor.com/compressor> (рисунок 1.3).

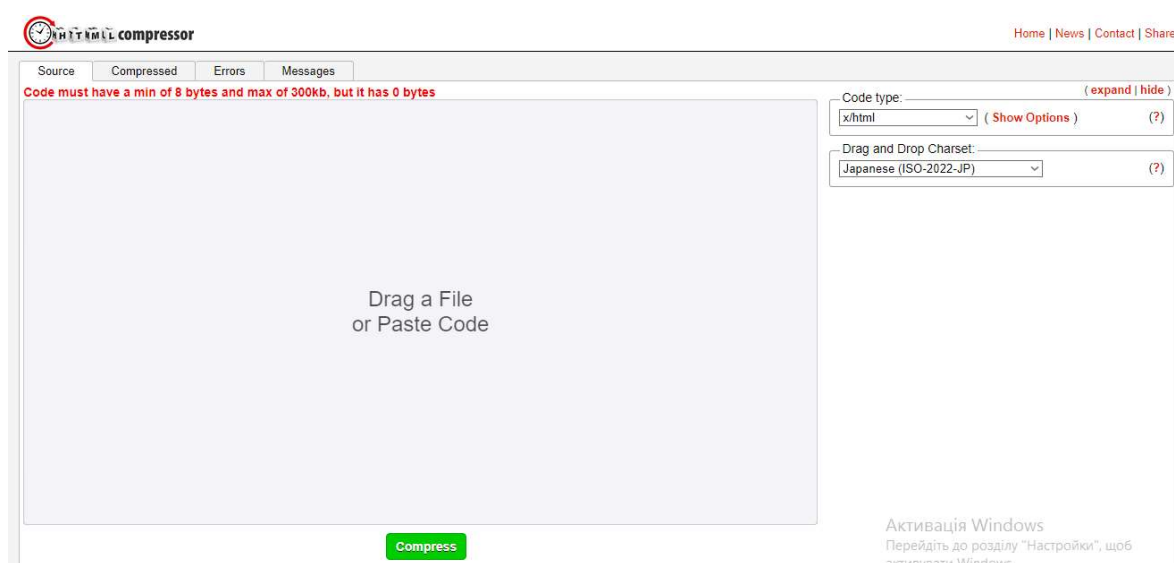


Рисунок 1.3

Відразу потрібно відмітити функціональність даного веб-ресурсу. На веб-сайті дана можливість вибирати кодування файлу, до якого збираємося застосувати компресію. Також є вибір компресії файлів для різних мов програмування і навіть їх поєднання. Є окреме відділення для помилок та повідомлень. Біля кожної незрозумілої кнопки присутні підказки. Проте сам інтерфейс дещо відлякує. Такий стиль використовувався у минулому і нових веб-ресурсів такого типу майже немає.

При детальнішому розгляді даного ресурсу стає зрозуміло, що тут розробники використали звичайну gzip компресію. Це дещо безглуздо, так як дана компресія вже відбувається на сервері. Завданням такого типу веб-

орієнтованих додатків являється більш мініфікація, вирізання лишньої частини коду.

Візьмемо наступний ресурс для прикладу – <http://mailfit.com/compressor/> (рисунок 1.4).

Веб-інтерфейс даної сторінки приємний і зрозумілий. Ресурс чудово виконує свою роботу. Єдина проблема, яка постійно відволікає це занадто велика кількість реклами. Вона дратує увагу користувачів.

Щодо самих результатів компресії даного веб-орієнтованого додатку є деякі проблеми. Розробники веб-сайту приділили достатню увагу видаленню пробілів різного типу в різних місцях, та інший «білих символів», та на жаль зовсім забули про пусті теги, значення атрибутів за замовчуванням і так далі.

Сжатие HTML, CSS, JS кода

Код для вставки

Результат

Вставьте CSS или JS код здесь...

Необходимо выбрать язык кода

☐ JS

☐ CSS

☐ HTML

Сжать

Очистить

Копировать

Рисунок 1.4

1.4. Обґрунтування теми дипломного проекту

На сьогодні компресія файлів різного типу це актуальна тема. Це економить не лише час, а й кошти. Питання економії ресурсів завжди гостро стоятиме перед людством.

Стиснення даних може значно зменшити обсяг зберігання файлу. У результаті стиснення адміністратори витрачають менше грошей і менше часу на зберігання.

Стиснення оптимізує продуктивність зберігання резервних копій і нещодавно виявилось в скороченні даних основного сховища. Стиснення буде важливим методом зменшення даних, оскільки дані продовжують рости в геометричній прогресії.

Практично будь-який тип файлу може бути стиснутий, але важливо дотримуватися найкращих практик при виборі, які з них стискати. Наприклад, деякі файли можуть вже бути стиснутими, тому стиснення цих файлів не матиме значного впливу.

За допомогою компресії HTML файлів ми зможемо пришвидчити завантаження сторінок веб-сайтів. Можемо досягти швидкого підключення від чого кількість клієнтів має збільшитися, так як при створення веб-сервісів потрібно орієнтуватися на проблеми клієнтів.

Стискання інакше відоме як кодування вмісту, - це публічно визначений спосіб стиснення текстового вмісту, перенесеного з веб-серверів у веб-переглядачі. Стиснення зазвичай використовує алгоритми стиснення у відкритому доступі, такі як gzip та compress, для стиснення HTML, JavaScript, CSS та інших текстових файлів на сервері. Цей стандартизований метод доставки стисненого контенту вбудований в HTTP 1.1, і більшість сучасних браузерів, які підтримують HTTP 1.1, підтримують ZLIB-інфляцію документів. Іншими словами, вони можуть автоматично розпаковувати стислі файли, що економить час і пропускну здатність.

Стандартні методи компресії на сервері – це лише кодування. Перед кодуванням потрібно ще виконати додаткову компресію або мініфікацію коду, так як кодування на сервері виконується з більшою частиною коду, який не несе ніякого технічного сенсу.

Наскільки швидко веб-сайт доступний для відвідувачів, впливає на показник відмов, досвід користувачів, розміщення в результатах пошуку (оскільки швидкість є фактором ранжування) та багато іншого. Насправді, швидкість завантаження настільки важлива, що пожна писати окремий дипломний проект на цю тему.

Одним з найбільш ефективних способів швидшого завантаження вашого сайту є стиснення HTML. Якщо ви регулярно працюєте з комп'ютером, ви, напевно, вже знаєте про стиск із архівів zip. Це відмінний спосіб зменшити розмір файлу, щоб полегшити їх перенесення. Те ж саме для HTML файлів, які потребують додаткової компресії перед стисненням.

Менші файли дозволять вашим відвідувачам завантажувати їх швидше і швидше переглядати ваш сайт. Це не тільки покращує користувацький досвід і робить сайт більш привабливим для пошукових систем, але також призводить до зменшення навантаження на сервер і використання пропускної здатності, що добре для пакетів хостингу на основі продуктивності.

Основна архітектура сайту, як всім, напевно, відомо, в найсучасніших веб-сайтах складається з поєднання HTML, CSS і JavaScript, які - на самому базовому рівні - не що інше, як текстові файли. Коли відвідувачі запитують побачити ваш сайт, їхні веб-переглядачі завантажують ці файли та роблять ваш сайт кодом, що міститься в ньому. Однак, оскільки код зазвичай обробляється людьми, він організований таким чином, що дозволяє легко зрозуміти для людських очей. Однак для таких машин, як ваш веб-переглядач, код просто не потрібний або нерозбірливий. Саме тому хороший спосіб швидшого завантаження веб-сайту - це конкатенація кодів і зменшення. Ці слова не означають нічого, крім вилучення всього форматування, яке є суто там, щоб зробити код більш розбірливим для людей, наприклад, коментарі, розриви рядків, пробіли та інші.

Просто в природі програмування і веб-дизайну є багато повторюваних шаблонів і елементів. Алгоритми стиснення, такі як Gzip, використовують цей факт, замінюючи повторювані візерунки на своєрідне скорочення.

2. АНАЛІЗ МОВ ПРОГРАМУВАННЯ ТА ТЕХНОЛОГІЙ РОЗРОБЛЕННЯ ВЕБ-ОРІЄНТОВАНИХ СИСТЕМ

2.1. Порівняльний аналіз мов програмування Python та Java Script

Вислів Джессі Фармера: «Мотивація, толерантність до розчарувань, соціальна підтримка та доступ до досвіду є більш важливими, ніж мова сировини, коли ви намагаєтеся стати програмістом».

При плануванні програмного рішення ви можете обрати багато різних мов програмування, і легко заблукати в тонкощах кожної з них. Ваш вибір мови може залежати від багатьох факторів. Якщо це для особистого проекту або хобі, ви можете погодитися на мову, яку ви знаєте. Якщо ваш вибір залежить від доступних ресурсів, ви можете отримати дійсно загадкові підходи. Або ж ви можете витратити багато часу на розробку багаторазових компонентів, що може призвести до того, що документація стане кошмаром.

Існує не тільки один фактор при обдумуванні вибору мови програмування. Наприклад, при розробці динамічної веб-сторінки ви можете розглядати JavaServer Pages (JSP) / сервлети як найкращий варіант, а інші можуть вважати за краще використовувати PHP або схожу мову сценаріїв. Жодна єдина мова не є «найкращим вибором». Хоча ви можете віддати перевагу

певним факторам, таким як продуктивність і безпека в корпоративних програмах, інші фактори, такі як менше рядків коду, можуть бути нижчими пріоритетами. Завжди є деякі компроміси.

Після того, як ви отримаєте проект або призначення, часто доводиться робити підготовчу роботу до вирішення актуальної проблеми.

Вибираючи мову для особистого проекту, ви можете обрати особистий фаворит. Тут важливі рядки коду; очевидним вибором є мова, яка може виконати роботу в 10 замість 20 рядків коду. Ви хочете спочатку отримати рішення, а потім піклуватися про акуратність або продуктивність.

Для проектів, створених для великої організації, це інший сценарій. Команди будуватимуть компоненти, які будуть взаємодіяти і з'єднуватися один з одним для вирішення певної проблеми. Вибір мови може включати такі фактори, як легкість перенесення програми на іншу платформу або наявність ресурсів.

Вибір потрібної мови програмування може забезпечити рішення, які є стислими, легко налагоджувати, легко розширювати, легко документувати та легко виправляти. Фактори, які необхідно враховувати при виборі мови програмування:

1. Цільова платформа
2. Еластичність мови
3. Час виробництва
4. Підтримка та спільнота

Найважливішим фактором, який необхідно враховувати, є платформа, на якій буде працювати програма. Подумайте з точки зору мови Java і C. Якщо програма написана на мові C і має працювати на машинах з операційними системами Windows і Linux, то це вимагатиме компіляторів платформ і двох різних виконуваних файлів. За допомогою мови Java, згенерований код байта достатній для запуску програми на будь-якій машині з інстальованою віртуальною машиною Java (JVM).

Подібний аргумент застосовується до веб-сайтів. Всі вони повинні виглядати і працювати однаково у всіх браузерах. Використовуючи теги CSS, HTML, не перевіряючи сумісність із веб-переглядачем, змусить один і той самий сайт виглядати і поводитися по-різному в браузерах.

"Еластичність" мови - це легкість, з якою нові функції можуть бути додані до існуючої програми. Еластичність може включати додавання нового набору функцій або використання існуючої бібліотеки для додавання нової функції.

Хоча порівняння цих мов не є технічно правильним, розглянемо Perl і Python. Perl підтримує вбудовані регулярні вирази як готову до використання функцію. У випадку Python, ви повинні імпортувати re модуль зі стандартної бібліотеки.

Час виробництва - це час, необхідний для того, щоб програма перейшла в життя - коли код готовий до виробництва і працюватиме так, як він призначений. Логіка презентації повинна бути додана до логіки керування при обчисленні часу виробництва.

Час виробництва дуже залежить від розміру коду. Теоретично, чим легше вивчати мову, тим менша кількість коду і, отже, менше часу для написання коду.

Наприклад, сайт управління контентом може бути розроблений за допомогою PHP-скриптів за декілька днів порівняно з кодом на Java, який може зайняти кілька місяців, якщо ви вивчаєте обидві мови з нуля.

Ви можете вичавити лише стільки продуктивності з програми та платформи скільки дозволить вибрана мова програмування. Мова, яка використовується для розробки програми, впливає на продуктивність. Є багато досліджень, які порівнюють швидкість мов програмування в одному середовищі. Ви можете бачити різні комп'ютерні тести для використання в якості посилання, хоча цифри не для конкретних оцінок продуктивності будь-якої мови.

Розглянемо веб-додаток, написаний як на Java, так і на Python. Дані про продуктивність, показані в тесті, приведуть вас до висновку, що, з огляду на подібні середовища, додаток, написаний на мові Java, має працювати швидше, ніж написаний на Python. А як щодо самого середовища? Якщо середовище є одним з основних версій Ubuntu Intel Q6600, це справедлива гра, оскільки обчислювальна потужність обмежена. Що робити, якщо веб-додаток знаходиться в хмарі, запущеному на Google App Engine? Тепер у вас є доступ до практично необмеженої обчислювальної потужності, і обидві програми повернуть результати майже в один і той же час. Фактор вибору тепер обертається навколо рядків коду та ремонтпридатності.

Продуктивність мови повинна мати значення, коли цільове середовище не дає великих можливостей для масштабування. Приклад такого середовища - ручні пристрої.

Так само, як хороше програмне забезпечення потребує співтовариства, що допомагає їй розвиватися, мова програмування також має сильну спільноту. Мова з активним форумом, ймовірно, буде більш популярною, ніж навіть чудова мова, яка не має своєї спільноти.

Спільна підтримка генерує вікіпедію, форуми, навчальні посібники та, що найважливіше, додаткові бібліотеки, які допомагають мові програмування зростати. Пройшли ті часи, коли люди працюють самотійно в замкнений приміщеннях. Розробники не хочуть перечитувати всю документацію, щоб вирішити одну незначну проблему. Якщо мова має достатню популярність, хороші шанси, що хтось ще зіткнувся зі своєю проблемою і написав про це у вікіпедії або форумі.

Perl - хороший приклад важливості спільноти. Всеохоплююча мережа Perl Archive (CPAN) - це зусилля громади. Основна мета CPAN - допомогти програмістам знайти модулі та програми, не включені до стандартного розподілу Perl. Його структура децентралізована; автори підтримують і вдосконалюють власні модулі.

За допомогою корпоративних додатків, дизайнери та програмісти повинні ходити по канату, коли мова йде про продуктивність, безпеку, ремонтпридатність і час розробки. Мова йде не тільки про використання мови програмування, що дозволяє отримати найкращі показники продуктивності. Інші важливі фактори включають: час на виробництво, еластичність і наскільки добре програма може інтегруватися в існуючу інфраструктуру.

Важливу роль відіграє і середовище, в якому буде використовуватися програма. Програми, написані на рівні підприємства, ніколи не є автономними. Кожна програма стає частиною ще більшої мети, тому інтерпретованість стає важливим чинником.

Уявіть собі, що підприємство з його веб-сервісами, реалізованими в кодї Java, хоче додати WebSphere MQ як надійну платформу. Немає сенсу використовувати API C WebSphere MQ для написання програми; вибір повинен бути кодом Java.

Припустимо, ваш наступний проект - це дослідження в областях, не пов'язаних з інформаційними технологіями та комп'ютерами. Наприклад, проект може включати обробку зображень, обробки звуку, водяні знаки або, можливо, дослідження фондового ринку. Необхідно створити код для імітації поведінки в реальному часі, але ви не є комп'ютером.

Проект вимагає багато швидкого і брудного коду. Найбільш актуальним фактором є час до виробництва. У цьому випадку час на виготовлення означає, як скоро ви можете зробити роботу компонента, щоб повернутися до великого завдання. Це дуже схоже на написання невеликих заглушок, не звертаючи уваги на взаємодію, на цьому етапі. Проект може стати повноцінним продуктом, але зараз він знаходиться на початкових етапах. Ваша основна вимога - створення прототипів.

Такі мови, як MATLAB і LISP, можуть прийти на допомогу. Якщо ви починаєте прототипування на C, ви будете вникати в подробиці змінних і

показчиків, не бачачи багато фактичного результату в умовах дослідження. MATLAB має інтеграцію з C / C ++ і Fortran, що дозволяє C-коду викликатися з MATLAB, і навпаки.

Існує кілька типів парадигм програмування. Основні чотири - це динамічний, статичний, інтерпретований та складений. Статично скомпільований і сильно типізований з одного боку або динамічний і інтерпретований, з іншого. Старі, традиційні мови програмування зазвичай вписуються в першу групу. Статично скомпільовані мови надають розробникам помилки під час компіляції, а не помилки під час виконання, забезпечуючи безпечніший, добре документований код з чистими інтерфейсом. Наприклад swift - це спроба об'єднати обидві можливості, і вона швидко зростає в популярності, проте дана мова в основному використовується при програмуванні під операційну систему Mac OS.

Також варто вивчити мови з нових парадигм для інтелектуального виклику, плюс це відмінний спосіб показати, наскільки добре налагоджені ваші навички.

Асамблер, це комп'ютерна програма-транслятор, що призначена для генерації машинного (двійкового) коду з символьного представлення цього коду (мови асемблера). До мови асемблера приводяться усі відомі мови.

C, оскільки він є спільним предком багатьох наступних мов і є мовою, на якій написано багато перекладачів і компіляторів цих мов.

Java це мова об'єктно-орієнтованого програмування та за допомогою якої виконується доступ до керованих середовищ виконання.

Haskell – це функціональна мова. Допомагає навчитися мислити функціонально; функціональне програмування, швидше за все, буде наступною основною парадигмою.

Коли починається новий проект, іноді важко вирішити, яка мова програмування буде найкращою для роботи. Хоча кожна мова програмування має свої сильні та слабкі сторони, часто існують причини, чому одна мова краща

за інші. Так само, як будівельник скаже вам, що набагато краще використовувати молоток, щоб забити цвях, ніж викрутку, досвідчений програміст скаже вам, що деякі мови краще інших, коли мова йде про різні завдання.

Перед тим як почати розробку веб-додатку потрібно вибрати найкомфортнішу технологію. Початок порівняльного аналізу пропоную почати з мов програмування. Для порівняння було вибрано найпопулярніші мови програмування, на даному відрізку часу, для створення веб-орієнтованих додатків. Це мови Python і Java Script.

Python - це мова з простим синтаксисом і потужним набором бібліотек. Це інтерпретована мова, з багатим середовищем програмування, включаючи надійний відладчик і профайлер. Python легка у вивченні мова програмування, проте широко використовується в багатьох наукових напрямках для дослідження даних.

Python існує в Інтернеті з 1995 року і є потужною мовою сценаріїв, що використовується для спільного "склеювання" програм. Простий синтаксис Python дозволяє легко вчитися і легко писати, оскільки розробники часто вибирають писати на Python, оскільки вони відчують себе більш продуктивними. Оскільки основна реалізація розробки для Python буде працювати в будь-якому середовищі, програмістам легко почати користуватися нею, незважаючи на свою операційну систему. Багато розробників звертаються до Python навіть, якщо до цього були незнайомі з нею і рідко знаходять, що це було поганим рішенням.

Хоча Python може бути швидким через його базову реалізацію C, вона може бути повільнішою, оскільки вона є вбудованою мовою. Накладні витрати на виконання можуть бути проблемою у великих додатках. Розгортання веб-додатків Python не настільки просте, як на інших мовах, як PHP, і веб-фреймворки Python ще не досягли корпоративного рівня, такого як Java або

.NET, коли йдеться про такі функції, як кластеризація, перехід на другий план і керування.

Javascript - це мова сценаріїв, яка в основному використовується в Інтернеті. Вона використовується для поліпшення HTML-сторінок і зазвичай знаходиться в HTML-коді. JavaScript є інтерпретованою мовою. Таким чином, її не потрібно компілювати. JavaScript надає веб-сторінці інтерактивності та динамічності. Це дозволяє сторінкам реагувати на події, демонструвати спеціальні ефекти, приймати змінний текст, перевіряти дані, створювати файли cookie, виявляти браузер користувача тощо.

Python можна використовувати для створення веб-додатків на стороні сервера. Хоча для створення веб-додатків не потрібна веб-структура, рідко розробники не використовують існуючі бібліотеки з відкритим кодом для прискорення прогресу в роботі своїх програм.

Python не використовується в веб-переглядачі. Мова, яка виконується в браузерах, таких як Chrome, Firefox і Internet Explorer, є JavaScript. Такі проекти, як rujs, можуть бути зібрані з Python на JavaScript. Однак більшість розробників Python записують свої веб-програми за допомогою комбінації Python і JavaScript. Python виконується на стороні сервера, а JavaScript завантажується на клієнт і запускається веб-браузером.

Також проблемою Python є однопоточність, що буде значно гальмувати роботу сервера. Звичайно, можна створити декілька потоків виконання, але в Java Script є кращий аналог – Event Loop.

Уже з цього короткого порівняльного аналізу зрозуміло, що кращим рішенням буде обрати Java Script як на клієнті так і на сервері. Завжди легше й зрозуміліше користуватися однією мовою програмування, особливо якщо написання й підтримка веб-орієнтованого додатку буде лежати на одній людині.

2.2. Аналіз реалізацій веб-архітектур

Веб-архітектуру можна визначити як концептуальну структуру інтернету. Типи веб-архітектури включають модель клієнт-сервер і трирівневу модель.

Веб-архітектура - концептуальна структура Всесвітньої павутини. WWW або Інтернет - це середовище, що постійно змінюється, що дозволяє здійснювати зв'язок між різними користувачами та технічну взаємодію (інтероперабельність) між різними системами та підсистемами. Основою для цього є різні компоненти та формати даних, які, як правило, розташовані в ярусах і будуються один на одного. Загалом, вони формують інфраструктуру Інтернету, що стає можливим завдяки трьом основним компонентам протоколів передачі даних (TCP / IP, HTTP, HTTPS), форматах подання (HTML, CSS, XML) і стандартам адресації (URI, URL). Термін веб-архітектура слід відрізняти від термінів архітектура сайту та інформаційної архітектури.

Всесвітня павутина - це концепція, яка була реалізована в 1990-х роках, щоб люди і машини могли спілкуватися один з одним у певному просторі. Він використовується для обміну, поширення та обміну інформацією в мережі. У той час веб складався переважно з статичних веб-сайтів на основі HTML, іншими словами, гіпертекстів, які можна отримати браузером. Пізніше були додані динамічні веб-сайти та розподілені веб-служби.

Інтернет - це середовище, яке постійно змінюється і розширюється багатьма розробниками, програмістами та різними консорціумами, такими як W3C. Однак використовувані архітектури можуть схематично відрізнятися.

Поговоримо про модель клієнт-сервер. Спочатку мережа складалася з дворівневої архітектури: клієнтів і серверів. Клієнти та сервери поділяли завдання та послуги, які система повинна була виконувати. Наприклад, клієнт може запитувати послугу від сервера; сервер відповідає на запит, надаючи послугу. Отримання веб-сайту за допомогою URL-адреси, яка спрямовує на

сервер для завантаження сайту в браузері клієнта, є прикладом двошарової моделі, також відомої як модель клієнт-сервер.

Сімейство інтернет-протоколів, яка наразі складається з близько 500 різних мережевих протоколів, зазвичай використовується як основа для WWW, але зазвичай вона містить еталонну модель TCP / TCP / IP. У веб-архітектурі повинні існувати три передумови, щоб розподілені прикладні системи могли взаємодіяти один з одним.

Формати подання з фіксованим стандартом: найбільш часто використовувані формати - HTML і CSS; або XML, коли машини спілкуються один з одним.

Протоколи для передачі даних: в Інтернеті використовується протокол HTTP (Hypertext Transfer Protocol) або HTTPS (Hypertext Transfer Protocol Secure). Інші програми, такі як поштові сервери, використовують протокол SMTP (Простий протокол передачі пошти) або POP (протокол пошти). Визначення використовуваних протоколів залежить від програми.

Стандарт для адресації: Це відноситься до URL (Uniform Resource Locator), який є прикладом більш загальної концепції URI.

Нарешті, веб-архітектура аналогічна операційній структурі прикладних систем для зберігання даних, передачі даних і презентації. Веб-архітектура, як правило, складається з серверів баз даних, які керують даними і ресурсами. Вони спілкуються з клієнтом за допомогою протоколу передачі, який може отримати дані та переглянути їх у браузері. Представлення зазвичай виконується за допомогою HTML і CSS.

Трирівневі моделі включають в себе логіку програми між клієнтом і сервером, яка обробляє обробку даних і дозволяє певну ступінь взаємодії. Наприклад, сервер додатків може обробляти дані, поки сервер баз даних призначений виключно для зберігання даних. Таким чином, вміст можна динамічно завантажувати і зберігати. Мова скрипту JavaScript часто відповідає за поведінку клієнта.

Як правило, розрізняють обробку даних на стороні сервера та на стороні клієнта. Динамічні веб-сайти характеризуються тим, що змінюється контент на стороні клієнта без необхідності нового зв'язку між сервером і клієнтом. Дія на стороні клієнта залежить від сценаріїв, тому асинхронна передача даних не потрібна. На стороні сервера змінений вміст зберігається через сервер додатків на сервері баз даних. За бажанням, це може бути віртуальний сервер, який емулює фізичний.

3. ОБҐРУНТУВАННЯ ВИБОРУ ЗАСОБІВ РЕАЛІЗАЦІЇ

3.1. Веб-платформа Node.js

Node.js - середовище виконання JavaScript. Робоче середовище Node включає в себе все, що потрібно для виконання програми, написаної на JavaScript.

Node.js з'явився, коли оригінальні розробники JavaScript розширили з того, що ви могли запускати лише в браузері, до того, що тепер це можна запускати працювати на вашому комп'ютері як окрему програму (рисунок 3.1). Тепер можна зробити більше з JavaScript, ніж просто розробляти інтерактивні веб-сайти.

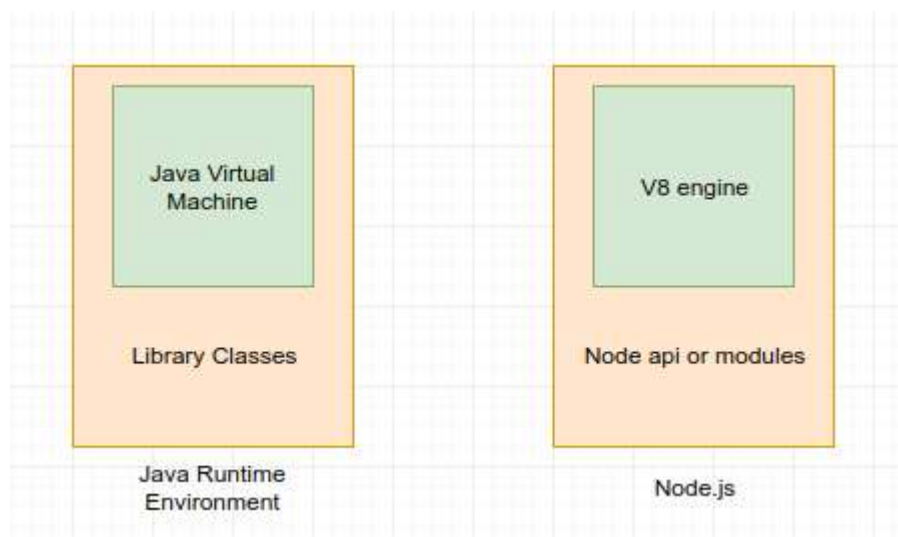


Рисунок 3.1

Тепер JavaScript має можливість робити те, що можуть робити інші мови сценаріїв, як Python. Обидва ваші браузері JavaScript і Node.js виконуються на V8 – це JavaScript двигун виконання написаний на мові програмування C компанією Google Chrome. Цей двигун приймає ваш код JavaScript і перетворює його в більш швидкий машинний код. Машинний код - це код низького рівня, який комп'ютер може виконувати без необхідності спочатку його інтерпретувати.

Нижче приведено формальне визначення, наведене на офіційному веб-сайті Node.js. Node.js® - це час виконання JavaScript, побудований на механізмі Chrome V8 JavaScript.

Node.js використовує керовану подіями неблокувальну модель введення-виведення, що робить його легким і ефективним. Екосистема пакета Node.js, прм, є найбільшою екосистемою бібліотек з відкритим кодом у світі.

Моделлю введення-виведення може бути що завгодно, починаючи від читання / запису локальних файлів до запиту HTTP до API. Дана модель займає час і, отже, блокує інші функції.

Розглянемо сценарій, в якому ми запитуємо базу даних backend для деталей про user1 і user2, а потім надрукуємо їх на екрані / консолі (рисунок 3.2). Відповідь на цей запит вимагає часу, але обидва запиту даних користувача можуть виконуватися незалежно і одночасно.

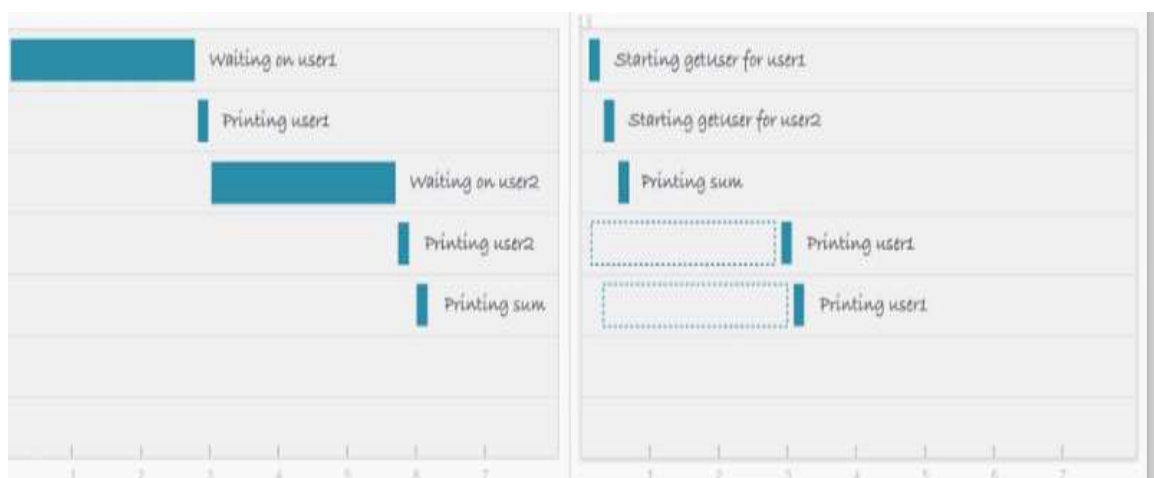


Рисунок 3.2

Спочатку розглянемо ситуацію блокування вводу-виводу. У методі блокування запит даних user2 не ініціюється, поки дані user1 не будуть надруковані на екран.

Якщо це був веб-сервер, нам потрібно було б почати новий потік для кожного нового користувача. Але JavaScript є однопоточним (точніше було б сказати він має однопоточний цикл подій, про який ми поговоримо трохи пізніше). Таким чином, JavaScript не дуже підходить для багатопоточних завдань.

Наступна ситуація з неблокуючим вводом / виводом. З іншого боку, використовуючи неблокуючий запит, можна ініціювати запит даних для user2, не чекаючи відповіді на запит на user1. Можна ініціювати обидва запити паралельно.

Ця модель неблокуючого вводу /виводу усуває необхідність багаторівневої обробки, оскільки сервер може одночасно обробляти декілька запитів.

Поговоримо про таке поняття як цикл подій JavaScript або event loop.

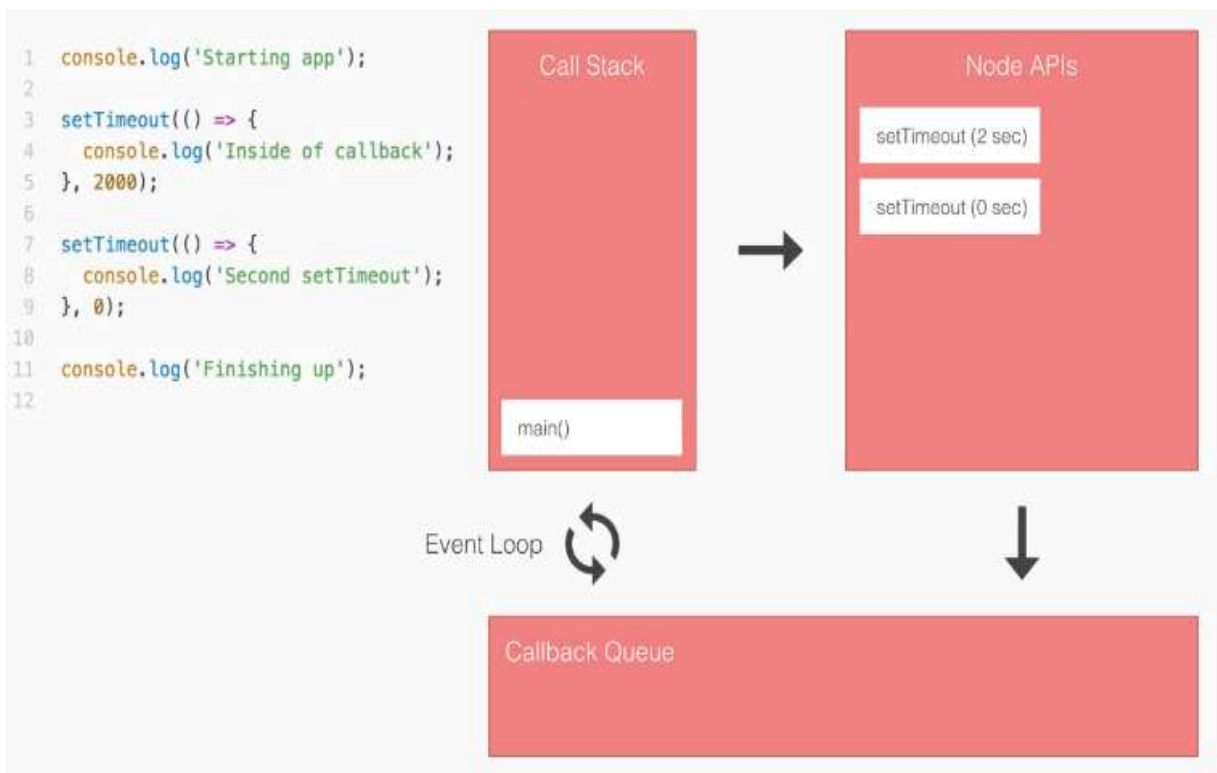


Рисунок 3.3

З наведеної схеми (рисунок 3.3) зрозуміло як працює Java Script і Node.js. Call stack даної технології є однопоточним, тобто одна дія виконується на одну одиницю часу. Всі паралельні процеси виконуються за допомогою двигунів написаних на C на схемі позначені як Node APIs. Усі вже виконані процеси попадають у чергу виконання (Callback Queue) де чекають. Сам же Event Loop відповідає за те щоб Call stack виконання не був пустим, беручи задачі для виконання з черги.

Таким чином ми розуміємо, що «однопоточний» Java Script може оброблювати успішно декілька процесів одночасно, що дає перевагу його над дійсно однопоточними мовами. Єдиною проблемою, яка лягає на плечі програмістів є уникати блокування Event Loop. Тобто старатися не залишати громіздкі обрахування в одному процесі, а розбивати їх на багато окремих асинхронних процесів.

3.2. Вибір шаблонізатора та фреймворка для верстки

Макет веб-сайту - це зразок, який визначає структуру веб-сайту. Він виконує функцію структурування інформації, присутньої на сайті, як для власника веб-сайту, так і для користувачів. Він забезпечує чіткі шляхи навігації в межах веб-сторінок і ставить найважливіші елементи веб-сайту та центру. Процес побудови макету і є так званою версткою.

Гарне компонування зберігає користувачів на сайті, оскільки це робить важливу інформацію легко та інтуїтивною доступною для пошуку. Погана схема розчаровує користувачів, які швидко залишають сайт, оскільки не можуть знайти те, що шукають.

З цієї причини краще витратити достатню кількість часу на знаходження гарного макету, оскільки користувачі не дадуть вам більше декількох секунд свого часу.

Між макетом і залученням користувачів до веб-сайту існує міцний зв'язок. Він визначає, як довго вони зупиняються на сторінках веб-сайту, скільки сторінок вони переглядають і як часто вони повертаються на сайт.

Отже, окрім подолання проблеми розбиття за секунду, хороша розкладка має додаткові переваги. Залучення відвідувачів може бути корисним.

При виборі макета може бути корисним також розглянути закон закриття. Він говорить, що, навіть якщо форма зображення не є повною, людське око прагне заповнити візуальні прогалини і розпізнати зображення в цілому. Як це може бути корисним для вас?

Ви не будете звертати увагу на деталі, а зосереджуєтесь на глобальному перегляді сторінок, що формують веб-сайт; користувачі знайдуть собі самотійно сенс-шлях.

Ви звертаєте увагу на деталі, використовуючи деякі додаткові секунди, щоб збільшити залучення експоненціально.

Якщо у вас є якийсь інтерес до веб-розробки, ви, напевно, чули про Bootstrap. Згідно з офіційним веб-сайтом, Bootstrap є найпопулярнішим HTML, CSS, і JS фреймворком для розробки адаптивних, мобільних перших проектів в Інтернеті.

Адаптивний дизайн увійшов у наше життя як обов'язкова очікувана норма. Для того щоб з легкістю швидко сформулювати свою сторінку в Bootstrap є всі для цього інструменти.

Bootstrap є вільною і відкритою версією фреймворка для створення веб-сайтів і веб-додатків. Структура Bootstrap побудована на HTML, CSS і JavaScript (JS), щоб полегшити розробку адаптивних, мобільних сайтів і програм.

Чуйний дизайн дозволяє веб-сторінці або програмі визначати розмір екрану та орієнтацію відвідувача, а також автоматично адаптувати дисплей відповідно. Перший підхід до мобільних пристроїв передбачає, що смартфони, планшети та мобільні додатки для конкретних завдань є основними

інструментами працівників для виконання робіт і вирішення вимог цих технологій в дизайні.

Bootstrap включає в себе компоненти інтерфейсу користувача, макети та інструменти JS, а також рамки для реалізації. Програмне забезпечення доступне попередньо скомпільоване або як вихідний код.

Марк Отто і Джейкоб Торнтон розробили Bootstrap на Twitter як засіб поліпшення узгодженості інструментів, що використовуються на сайті, і зменшення технічного обслуговування. Програмне забезпечення раніше називалося Twitter Blueprint і іноді називається Twitter Bootstrap.

У комп'ютерах слово bootstrap означає завантажувати: завантажувати програму на комп'ютер за допомогою набагато меншої початкової програми для завантаження в потрібну програму (яка зазвичай є операційною системою).

У фізичному світі завантажувальний пристрій - це невеликий ремінець або петля на задній частині шкіряного завантаження, що дозволяє витягувати весь завантажувач і в загальному використанні, завантажувальний засіб - це використання невеликих початкових зусиль у щось більше і більше. Існує також загальний вираз, "підтягуючи себе власними підніжками", маючи на увазі, що вони зможуть досягти успіху з невеликого початку.

Тепер обговоримо шаблонізатор. Оскільки HTML як ми знаємо складається з кучі повторів однотипного тексту, то часто для швидкого формування сторінки використовують шаблонізатор. Це значно економить час, що являється найціннішим ресурсом розробників. Для веб-орієнтованого додатку для компресії HTML файлів було вибрано шаблонізатор Pug. Це зручна бібліотека яка входить до стандартного менеджера пакетів Node.js.

Отже маючи в запасі фреймворк для HTML – Bootstrap, та шаблонізатор Pug ми можемо створити приємний сучасний інтерфейс, в якому користувачам буде максимально комфортно у користуванні.

3.3. Вибір засобів хостингу веб-додатку

«Великий хостинг зводиться до 3 пунктів: швидкість, підтримка і безпека», сказав Адам Беррі, цифровий директор Wingard Creative. Симон Болл, керівник відділу цифрових медіа в Nasdaq Corporate Solutions, заявив, що масштабованість є критичною. «Вам потрібна можливість швидко масштабувати ваш веб-сайт, коли ваша цільова аудиторія зростає і стійкість до обробки раптових вибухів високого трафіку», - сказав Бал.

Послуги хостингу доступні в широкому діапазоні цін від кількох доларів на місяць до тисяч доларів. Якщо ви починаєте малий бізнес, ви, ймовірно, можете добре погодитися з хмарою, віртуальним приватним сервером або керованим сервісом від \$ 10 до \$ 100 (приблизно £ 8 до £ 80 або AU \$ 15 до \$ 130) на місяць.

Звернемо увагу на те, що експерти рекомендують враховувати при виборі хостинг-провайдера і посилання на різні списки послуг і знижок CNET Commerce. CNET може отримати частку доходу від продажу цих продуктів і послуг, пропонованих нашими партнерами.

Вирішіть, скільки рук потрібно. Базове обслуговування клієнтів забезпечує доступ до електронної пошти, квитка та підтримки телефону. Проте час виконання запитів буде різним. Деякі постачальники послуг навіть пропонують цілодобову телефонну підтримку. Обмежуючим фактором для некерованого сервісу є те, що в той час, коли постачальник може відповісти на питання про базову конфігурацію, він не буде вашим системним менеджером.

Якщо ви хочете повністю делегувати управління своїм сайтом, ви хочете розглянути керовану службу. Постачальники керованого сервісу переконуються, що ваша система налаштована належним чином для вашого навантаження, стежити за проблемами безпеки, виправляти їх буде ваше програмне забезпечення в міру необхідності та керувати резервними копіями серед інших завдань.

Оцініть обсяг трафіку, який ви очікуєте (і будьте чесними до себе). Хостинг-провайдери зазвичай стягують плату на основі зберігання та використання пропускної здатності. Пропускна здатність - це показник кількості байтів, які ви обслуговуєте за певний період. Якщо ви очікуєте, що лише кілька людей відвідають ваш сайт, пропускна здатність буде низькою. Але якщо ви раптом з'явилися у верхній частині Google або ваш продукт стає вірусним, ви можете очікувати, що вимоги до пропускної здатності збільшаться.

До тих пір, поки ви чесні з самим собою, ризику немає. Наприклад, якщо ви плануєте обслуговувати лише декілька сторінок для кількох місцевих клієнтів, ви ніколи не зіштовхнетеся з будь-якими обмеженнями. Але якщо ви знаєте, що ви дійсно створюєте сайт, який підкреслюватиме низькі рівні спільних серверів, не забудьте обрати виділений або хмарний сервер. Це наступне.

Розуміти типи серверів. Найдешевший хостинг доступний на спільних серверах, де в одному вікні можуть працювати сотні веб-сайтів. Продуктивність вашого сайту залежить від навантаження на всі інші сайти, що розміщуються на хості. Спільний хостинг також обмежує доступ до можливостей сервера, що, як правило, обмежує завантаження файлів через FTP або SFTP, запобігаючи доступу до оболонки, обмежуючи, які програми можна запускати на сервері та обмежуючи доступ до бази даних, якою може користуватися ваш сайт.

Наступним ярусом є VPS (для віртуального приватного сервера), який є повним екземпляром віртуальної машини (модельованого комп'ютера), що працює на коробці. Як правило, хостинг-провайдери запускають багато екземплярів VPS на одному полі, але продуктивність майже завжди краще, ніж спільні служби базового рівня. Якщо ви використовуєте VPS, ви повинні бути знайомі з базовим обслуговуванням і керуванням сервером.

Якщо ви не хочете поділяти продуктивність з іншими сайтами, розгляньте виділений сервер, фізичний ящик, який вам здається в оренду. Це те ж саме, що сервер, що сидить за столом, за винятком того, що він розташований у центрі обробки даних постачальника послуг. Застосовуються лише ті, хто володіє навичками управління системою.

Cloud-сервери можуть бути кращим вибором. Вони зазвичай працюють на гігантських публічних хмарах, таких як Amazon Web Services або Microsoft Azure. Постачальники послуг можуть будувати будь-яку конфігурацію, яка відповідає потребам своїх клієнтів. Великою перевагою хмарних серверів є те, що ви можете легко масштабувати. Якщо вам потрібно мати змогу впоратися з цим великим стрибком трафіку, просто сплачуйте своєму провайдеру більше грошей. Нічого не потрібно переміщати або перебудовувати.

Будьте обережні з необмеженими пропозиціями. Деякі хостинг-провайдери пропонують так зване необмежене сховище і пропускну здатність на кілька доларів на місяць. Ця угода часто не є такою, якою вона здається. Якщо ви платите три долари на місяць за хостинг, то, ймовірно, буде щось у ваших умовах обслуговування, що дозволяє вашому хостинг-провайдеру або придушити вашу продуктивність, або закрити вас після певного рівня використання.

Виберіть портативну систему керування вмістом, щоб уникнути блокування. Більше хостів це досить добре, але час змінюється. Зміни в управлінні, придбання та технологічні зрушення можуть змінити ваш веб-хостинг. Переконайтеся, що ваш сайт не заблоковано для жодного хосту, і що у вас є практика резервного копіювання.

Часто для бізнесу, потрібно переконуватись, що ви використовуєте відкриту систему управління контентом. Багато людей використовують WordPress поверх PHP, який буде працювати на будь-якому рівні. Здійснюйте регулярні оновлення та резервні копії сайтів, і тоді завжди матимете доступ до

даних, засобів масової інформації та структури вашого сайту. Цей підхід демонструє все, що потрібно.

Кожен веб-сайт потребує веб-хостингу - компанії, яка надає серверам, які зберігають і живлять ваш сайт. Хостинг, як правило, пропонується через реєстратора доменів, і ви можете мати кілька варіантів сортування. Перш ніж зробити вибір, потрібно розглянути чотири простих кроки, які допоможуть визначити, який веб-хост найбільше підходить:

1. Потрібно з'ясувати потреби сайту. Що ви будете робити зі своїм веб-сайтом? Правильний веб-хост буде підтримувати платформи або будівельні інструменти, за допомогою яких створюється потрібний сайт, а також забезпечити ваш сайт адекватним сховищем, швидкістю та надійністю. Наприклад, потреба обробляти онлайн-платежі. Майбутній веб-орієнтований додаток буде блогом або візуальною галереєю. Деякі веб-хости пропонують інструменти для створення веб-сайтів та інфраструктуру, яка спеціалізується на певних типах веб-сайтів або функцій. Той, що працює для когось іншого, може не підходити для вас.

2. Розгляньте варіанти створення веб-сторінок і пакетної комплектації. Ви створюєте свій власний сайт, або бажаєте використовувати веб-будівельника DІY? Є кілька варіантів того, як ви хочете, щоб ваші послуги були в комплекті. Веб-хост і будівельник об'єднані: багато веб-будівельних послуг також пропонують хостинг. Це дозволяє створювати власний веб-сайт і отримувати хостинг як частину пакета. Прикладами веб-хостів, які також включають будівельників, є Wix, Weebly, Squarespace, Shopify і Blogger. Домени Google співпрацюють з усіма об'єднаними веб-хостами та будівельниками. Читайте більше в документації до окремих сервісів які надають послуги хостингу.

Ви також можете вибрати веб-хост окремо від будівельника. Тобто веб-хост, який не включає веб-будівництво, якщо ви створюєте свій власний веб-сайт або використовуєте конкретного будівельника, який не включає хостинг, наприклад, WordPress. (Ви можете використовувати інструмент WordPress для

публікації вашого сайту на власному хості або використовувати власну службу хостингу, Bluehost).

3. Зробіть деякі дослідження. Як тільки ви знаєте, що вам потрібно для вашого сайту і які послуги ви шукаєте, прийшов час для вивчення веб-хостингу і будівельника варіантів. Є кілька поглиблених статей, які можуть допомогти вам знайти і порівняти різних хостів і будівельників (подивіться на webhostingsecrets розкриті, thesitewizard, веб-дизайн і наш пост про те, як вибрати веб-будівельника), або ви можете зробити деякі пошуки по собі. Складіть короткий перелік основних варіантів. Подумайте про потреби вашого сайту зараз і в майбутньому.

4. Порівняйте ключові фактори. Конкретні речі, які слід шукати у веб-хості та будівельнику сайтів, можуть змінюватись залежно від ваших потреб, тому використовуйте список нижче, щоб з'ясувати, що найбільше для вас.

Важливо цважно переглядати ціну реєстрації та оновлення на додаток до вартості реєстрації, переконайтеся, що ви перевірили ставки поновлення та перечитали весь дрібний шрифт за будь-які приховані платежі або зобов'язання.

Технічні потреби. Переконайтеся, що вибраний веб-хост і будівельник сайтів відповідає всім вимогам вашого сайту, а також перевірте, що ви захочете або стане потрібним в майбутньому, наприклад: більша кількість доменів, які можна розмістити, простір сервера, швидкість і надійність, обмеження пропускної здатності та використання.

Налаштування управління та простота використання. Подумайте, хто буде робити основну частину управління сайтом - наприклад, оновлення вмісту сайту.

Після того як ви знайдете веб-хост і будівельника, який відповідає вашим потребам, ви будете на один крок ближче до створення вашого сайту. Якщо ви готові приступити до роботи, отримайте відмінне ім'я домену та безліч варіантів хостингу з доменів Google.ч

Для того щоб мати можливість демонструвати роботу веб-орієнтованого додатку для компресії HTML файлів потрібно обрати платформу для хостингу. Найпопулярнішими є Digital Ocean і Heroku.

Digital Ocean надають окремий сервер. Можна вибрати власний домен. Нададуть достатню кількість пам'яті, та необмежену кількість користувачів на сайті. Знову ж таки всі характеристики напряду залежитимуть від плану що ми виберемо. Також даний сервіс хостингу є достатньо надійний, підтвердженням цьому факту є їх клієнтська база, а це більшість відомих сайтів. Проте найдешевший їх тарифний план коштуватиме від 5\$ на місяць.

Безкоштовним аналогом для демонстрацій можна вибрати Heroku. У цьому сервісі для хостингу є безкоштовна можливість хостити свій веб-орієнтований додаток для компресії HTML файлів. Про те є деякі мінуси. Наприклад прямого доступу до серверу не буде. Все встановлення програмного забезпечення буде проходити через спеціальні команди Heroku. Тому потрібно відповідально до цього віднестися. Про те варто згадати що ми вибрали Java Script Node.js, а в цій платформі хостинг хоч на сервісі Heroku, хоч на окремому сервері дуже простий, особливо якщо порівнювати з мовою програмування Python і її веб-фреймворком Django. Також варто зазначити що на даному веб-сервісі для хостингу в безкоштовному тарифному плані є обмеження на кількість одночасних користувачів у веб-орієнтованому додатку, який хоститься. Також існує обмеження на кількість днів. І в доповнення до всього вище названого домен виглядатиме як <https://htmlcompressor.herokuapp.com/>.

Після порівняння було вибрано сервіс для хостингу Heroku. Так як його безкоштовних можливостей вистачає для демонстраційного хостингу. Також було вибрано домен для веб-орієнтованого додатку для компресії HTML файлів - <https://htmlcompressor.herokuapp.com/>.

4. РОЗРОБКА ВЕБ-ОРІЄНТОВАНОГО ДОДАТКУ ДЛЯ

КОМПРЕСІЇ HTML ФАЙЛІВ

4.1. Інтерфейс веб-орієнтованого додатку

Під час аналізу аналогічних веб-орієнтованих додатків для компресії HTML файлів було прийнято рішення розробити максимально простий і зрозумілий інтерфейс. Ніщо не має відволікати увагу від основної задачі сторінки. Кольорова гама інтерфейсу веб-орієнтованого додатку також має бути максимально скромною.

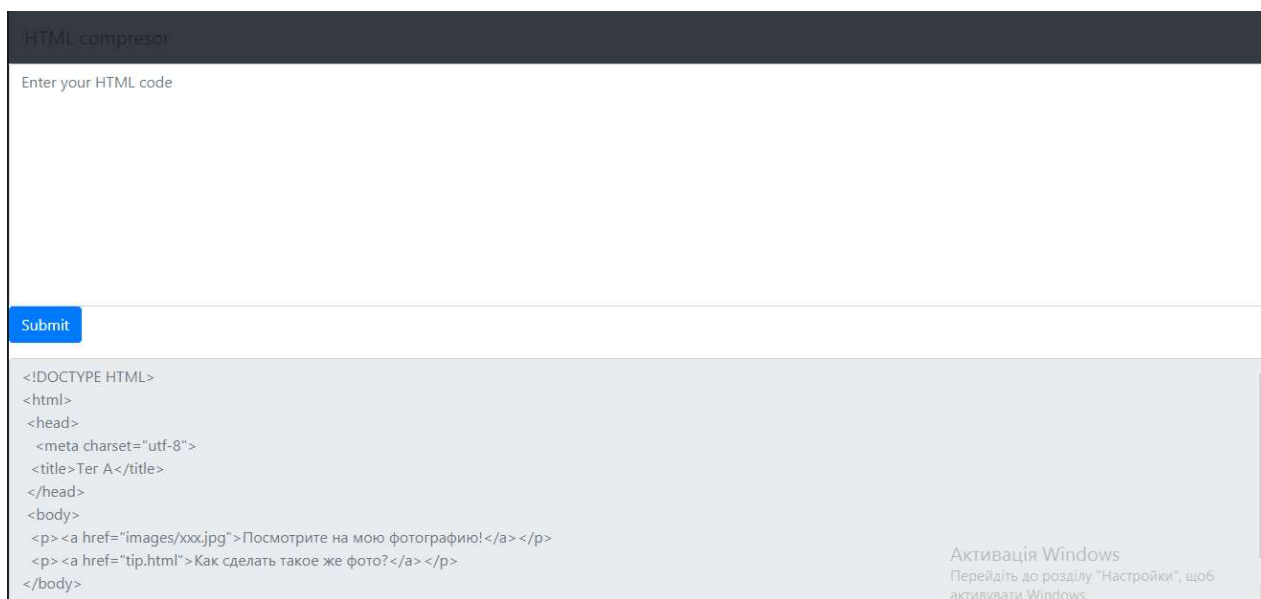


Рисунок 4.1

Зверху екрану розміститься навібар з назвою сторінки (рисунок 4.1). Далі екран буде умовно поділений на дві частини вхідного і вихідного коду. Вже скомпресований код не можна буде змінювати на сторінці, задля того щоб користувачі не могли нашкодити собі внісши непоправні помилки у скомпресований код.

У полі вводу знаходиться зрозумілий посил, що саме туди має бути введено клієнтський HTML код «Enter your HTML code». І доки код не скомпресовано поле з виводом взагалі ніяк не відображатиметься на екрані. Це також для кращої зрозумілості, щоб якщо користувач вперше заходить на веб-сторінку не плутався у полях.

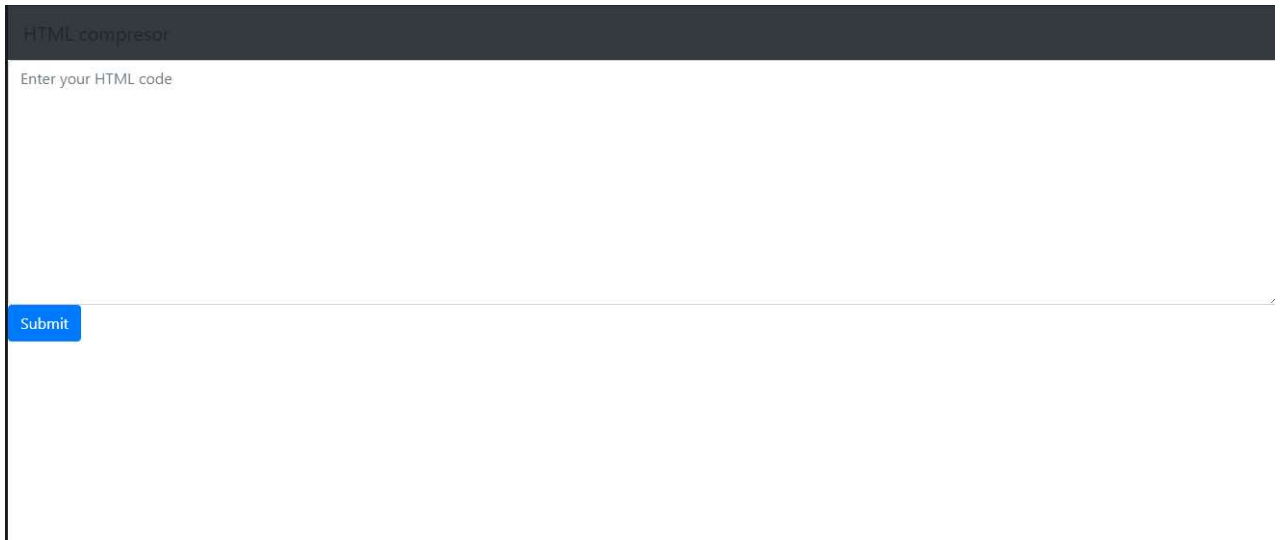


Рисунок 4.2

Кнопка «Submit» синього кольору, для того щоб відрізнитися і привертати на себе увагу користувачів (рисунок 4.2). Після натискання на кнопку введений код компресується і виводиться у нижній відділ екрану, а верхнє поле автоматично очищується, готуючись до прийому нового HTML коду.

4.2. Структура проекту веб-орієнтованого додатку

Перш ніж приступити до створення веб-сайту, слід подумати над організацією файлів і папок нашої веб-сторінки. Ми могли б скинути все в одну папку, і наш HTML-код не заперечував би, доки наші шляхи до файлів були налаштовані правильно, але це дуже швидко нас заплутає. Ми можемо легко створити чисте, організоване дерево файлів для нашого сайту, що не тільки полегшить наше життя, але й покращить досвід наших відвідувачів. Ми будемо використовувати графічний інтерфейс користувача для керування файлами та папками для створення нашого файлового дерева. Ваша операційна система майже напевно поставляється з одним таким менеджером файлів. В даному проекті було використано Nemo, оскільки він приголомшливий, і він є менеджером файлів за замовчуванням для Linux Mint. Якщо ви працюєте у Windows, ви, швидше за все, будете використовувати Провідник Windows, а користувачі Mac OS матимуть програму під назвою Finder. Процес майже

однаковий незалежно від вашого браузера файлів, тому ви не повинні мати жодних проблем зі мною.

Дуже важливо для подальшої долі проекту правильно вибрати його структуру. Це полегшить не лише написання коду, але і його відлагодження, та в подальшому - підтримку. Часто розробники приділяють надто мало уваги цьому аспекту.

Згідно з схеми основною папкою проекту є папка «HTMLcompressor» в ній лежать всі дочірні папки (рисунки 4.3). Файли в корні проекту залишаються лише ті, які потрібні для запуску проекту.

Далі розташована папка «config» в ній розташовуються конфігураційні файли з потрібними для запуску, наприклад у девелоперському середовищі, константами.

У папці «node_modules» розташовані всі додаткові зовнішні модулі або бібліотеки.

Сам код і всі написані розробником модулі знаходяться в «src». Вона поділяється на «router», де знаходиться код який оброблює запити, які приходять на шляхи веб-орієнтованого додатку для компресії HTML файлів.

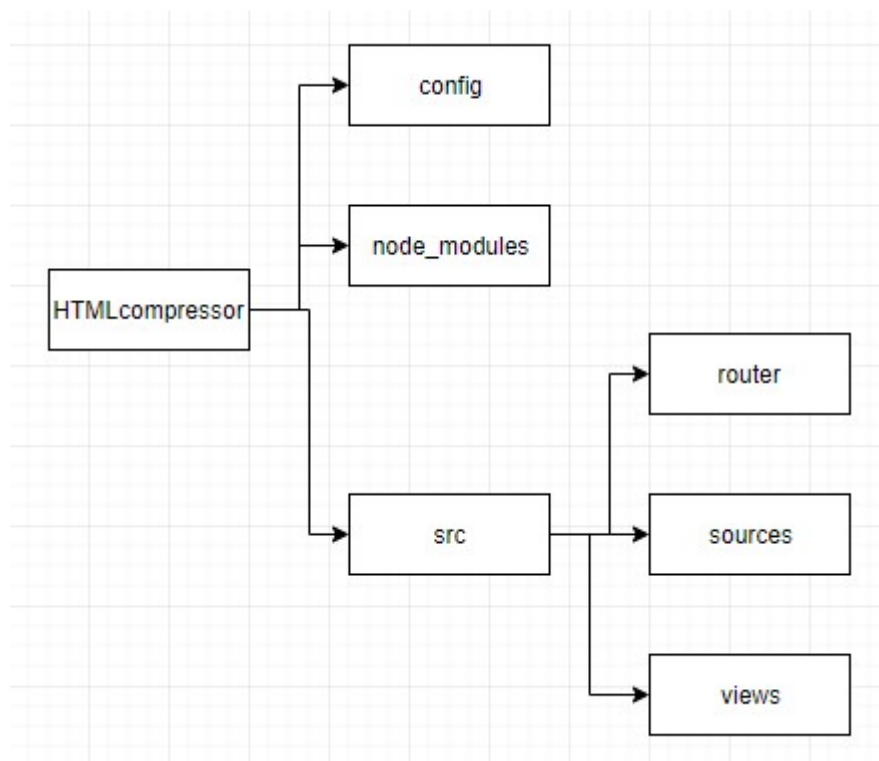


Рисунок 4.3

У «sources» знаходиться сам алгоритм компресії HTML файлів і деякі допоміжні модулі, які написані спеціально під даний проект. І нарешті всі шаблони веб-сторінки, створені за допомогою шаблонізатора, знаходяться у «views».

Дана представлена модель структури проекту є більш загальною прийнятою розробниками і нею користуються у великих проектах. Достатньо зрозуміле розташування дає постійне розуміння того, де що знаходиться якщо потрібно. Також ще дуже важливо правильно називати самі файли. Для того щоб завжди мати розуміння у вашому проекті, де що знаходиться.

4.3. Алгоритм компресії HTML файлів

Ми вже аналізували важливість обслуговування веб-сторінок користувачів швидким часом завантаження. Погана веб-продуктивність дорівнює втраченим доходам і клієнтам. Якщо ви продовжуватимете створювати та покращуватиме ефективність веб-сайту, потрібно шукати

геніальні способи витиснення цих дорогоцінних мілісекунд із часу завантаження сторінки.

Для перевірки продуктивності веб-орієнтованих додатків можна використати плагін Google PageSpeed, оскільки він надає корисні поради для покращення загальної ефективності та часу завантаження сторінки.

Зменшення HTML-сторінки - це процес видалення непотрібних символів або пробілів з веб-сторінки. Це не змінює його відображення або функціональність, а також покращує загальну продуктивність веб-сторінки. Процес мінімізації HTML подібний до мінімізації / ущільнення CSS, але можуть бути випадки, коли ви не повинні стискати HTML, щоб зберегти цільність HTML-коду.

Центральною причиною мінімізації HTML є підвищення продуктивності за рахунок видалення непотрібних символів і пробілів у файлі HTML. Коли ви видаляєте символи та пробіли, ви зменшуєте загальний файл. Це, в свою чергу, полегшує завантаження веб-сервера. Одним з основних прикладів символів, які можна видалити, є символи, які становлять коментар. Вважається доцільною надання коментарів у коді документації. Зробіть копію файлу та збережіть його в папці резервного копіювання або папці з позначкою "документація", щоб зберегти коментарі, які містяться у вашому коді. Змініть файл, який буде завантажено на сервер. Це допоможе підвищити його продуктивність.

Якщо ви виявите, що ваші веб-сторінки постійно завантажуються повільно, то одним із кроків, які можна виконати, є перегляд HTML-коду, який складається з вашого веб-сайту. Видаліть непотрібні символи, пробіли або коментарі, щоб покращити час завантаження веб-сайтів. Видалення цих елементів не повинно впливати на функцію або відображення вашого веб-сайту. Зменшення HTML-коду має бути лише вашою метою оптимізації ефективності веб-сайту, оскільки це може не бути важливим чинником для швидшого завантаження веб-сайту. Оптимізація продуктивності веб-сайту може також

включати зменшення кількості CSS, налаштування кешування або інших дій на стороні сервера.

Існують певні випадки, коли видалення пробілів може спричинити проблеми в елементах, які ви намагаєтеся відобразити. Хорошим прикладом є те, що ви використовуєте тег `<pre>`, де ви намагаєтеся представити блок коду певним чином. У цих випадках ви уникнете видалення пробілів, щоб елементи, які ви намагаєтеся відобразити, відображалися належним чином. Якщо виявиться, що мінімізація HTML впливає на спосіб відображення веб-сторінки, переконайтеся, що не стискаєте цей розділ коду.

Сама мова розмітки HTML має деревовидну структуру, тому для аналізу вона достатньо зручна. Розглянемо роботу реалізованого алгоритму компресії HTML файлів у веб-орієнтованому додатку.

Перш за все при аналізі надійшовшого HTML файлу ми видаляємо пробіли між тегами. Насправді звучить смішно, але такі пробіли або знаки табуляції, які ми використовуємо для формування структури коду і не несуть ніякого позитивного навантаження, можуть займати достатньо багато місця. Переноси між рядками між тегами, теж підпадають під категорію непотрібних, тому видаляються. Також видаляємо пробіли, які знаходяться між елементами, між яким уже присутні розділові знаки. Наприклад з «display: inline» зробимо «display:inline». Наступним кроком пробіли поміж елементів між якими відсутні розділові знаки згортаємо до одного. Також часта ситуація, коли ми навіть не помічаємо скільки пробілів де поставили. Такі ситуації дуже важливі для мінімізації або компресії, коли кожен байт важливий.

У випадку помилки виявленої у HTML файлі алгоритм не буде випадати з помилкою, а максимально оптимізує ті дані, які в нього є. Це зроблено з оглядом на те, що даний веб-орієнтований додаток не являється компілятором і відшук та розгляд помилок не входить в його компетенцію.

Видаляємо коментарі, вони потрібні лише розробникам, для легшого читання коду. І зовсім не потрібні при його виконанні. Видаляємо лапки, якщо

вони не потрібні. Видаляємо всі пусті атрибути в тегах, атрибути із значеннями по дефолту. Видаленню також підлягають всі пусті елементи. Ще непотрібні такі атрибути як «type="text/javascript"», або «type="text/css"» в тегах стилю або посилань.

Також варто зауважити, що даний веб-орієнтований додаток для компресії HTML файлів створений для мови розмітки HTML п'ятої версії. Тому також за алгоритмом ми заміняємо довгий заголовок HTML на більш короткий прийнятий в п'ятій версії.

					ІАЛЦ.467200.004 ПЗ	Лист
Зм	Лист	№ докум.	Підп.	Дата		47

ВИСНОВОК

Темою даного диплому було вибрано створення веб-орієнтованого додатку для компресії HTML файлів. Компресія у сучасному світі дуже гостро й проблематичне питання, так як світові ресурси сховищ постійно зменшуються, а об'єми даних які зберігаються постійно збільшуються. У веб-індустрії компресія також має значну роль. За допомогою компресії HTML ми не лише зможемо зменшити об'єми використовуваної пам'яті, а й збільшимо швидкість завантаження сторінки.

Було описано аналіз вже існуючих веб-орієнтованих додатків для компресії HTML файлів. Інші існуючі аналоги були дуже близькими, або майже ідентичні до описаних.

Аналізуючи вже існуючі веб-орієнтовані додатки було створено власний компресор HTML файлів. Він враховує всі функціональні потреби, має простий та швидкий інтерфейс, а також показує відсоток на який відбулася компресія в порівнянні з вхідним файлом.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Стаття про дослідження компресії в комп'ютерних технологіях [Електронний ресурс]. – 2019. – Режим доступу: <https://techxplore.com/news/2019-04-data-compression-technique-faster.html>. Дата доступу: травень 2019.
2. Дослідження популярності мов програмування [Електронний ресурс]. – 2018. – Режим доступу: <https://intersog.com/blog/tech-tips/choosing-the-right-programming-languages-for-your-development-projects>. – Дата доступу: березень 2019.
3. Стаття про вибір методів розробки [Електронний ресурс]. – 2017. – Режим доступу: <https://www.ibm.com/developerworks/library/wa-optimal/index.html>. – Дата доступу: квітень 2019.
4. Стаття порад для вибору сервісу для хостингу [Електронний ресурс]. – 2019. – Режим доступу: <https://www.cnet.com/features/amazons-helping-police-build-a-surveillance-network-with-ring-doorbells/>. - Дата доступу: травень 2019.
5. Поради для вибору сервісу для хостингу [Електронний ресурс]. – 2016. – Режим доступу: https://domains.google/intl/en_in/learn/how-to-choose-a-web-host-in-4-steps.html#/. Дата доступу: травень 2019.
6. Andrew B. King Website Optimization: Speed, Search Engine & Conversion Rate Secrets [Текст] – стор 247-290.
7. Стаття про мінімізацію HTML [Електронний ресурс]. – 2016. – Режим доступу: <https://www.inmotionhosting.com/support/edu/website-design/minifying-html>. Дата доступу: березень 2019.
8. Стаття про організацію веб проектів [Електронний ресурс]. – 2014. – Режим доступу: <https://thehelloworldprogram.com/web-development/creating-files-folder-structure-web-pages/>. Дата доступу: квітень 2019.